2009-03

Inexpensive solutions for direction finding of bridge-to-bridge radio transmitters using phase difference in received signal

Boernke, Eric Paul.

Monterey, California. Naval Postgraduate School

NAVAL POSTGRADUATE SCHOOL

MONTEREY, CALIFORNIA

THESIS

INEXPENSIVE SOLUTIONS FOR DIRECTION FINDING OF BRIDGE-TO-BRIDGE RADIO TRANSMITTERS USING PHASE DIFFERENCE IN RECEIVED SIGNAL

by

Eric Paul Boernke

March 2009

Thesis Advisor:	Frank E. Kragh
Co-Advisor:	Herschel H. Loomis

Approved for public release; distribution is unlimited

THIS PAGE INTENTIONALLY LEFT BLANK

REPORT DOCUMENTATION PAGE			Form Approved OMB No. 0704-0188
Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instruction, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188) Washington DC 20503.			
1. AGENCY USE ONLY *(Leave blank)*	**2. REPORT DATE** March 2009	**3. REPORT TYPE AND DATES COVERED** Master's Thesis	
4. TITLE AND SUBTITLE Inexpensive Solutions for Direction Finding of Bridge-to-Bridge Radio Transmitters Using Phase Difference in Received Signal		**5. FUNDING NUMBERS**	
6. AUTHOR(S) Eric Paul Boernke			
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) Naval Postgraduate School Monterey, CA 93943-5000		**8. PERFORMING ORGANIZATION REPORT NUMBER**	
9. SPONSORING /MONITORING AGENCY NAME(S) AND ADDRESS(ES) N/A		**10. SPONSORING/MONITORING AGENCY REPORT NUMBER**	
11. SUPPLEMENTARY NOTES The views expressed in this thesis are those of the author and do not reflect the official policy or position of the Department of Defense or the U.S. Government.			
12a. DISTRIBUTION / AVAILABILITY STATEMENT Approved for public release; distribution is unlimited.		**12b. DISTRIBUTION CODE**	

13. ABSTRACT (maximum 200 words)

This thesis presents two methods for determining the bearing of a source generating a Very High Frequency (VHF) bridge-to-bridge radio transmission. Using principles of interferometry, one can utilize the calculated phase difference between the received signal at multiple antennas to determine the Angle of Arrival of the detected transmission. This translation of phase difference to Angle of Arrival is accomplished using equations based on signal properties and geometry. The theoretical method is shown proving the relationship between Angle of Arrival and phase difference, as well as how a single platform could accomplish this detection and calculation. Theoretical simulation was accomplished using various simulation tools including Mathwork's Simulink and Tonne Software's Elsie. Methods are then provided to detect the phase difference using both a series of analog mixers and filters as well as digitally, using software radio. Analog filters were built and tested to determine the relationship between phase difference and voltage output. Software programs were written for a Software-Defined Radio implementing digital filtering to verify the analog performance. Results and accuracy are shown based on initial testing.

14. SUBJECT TERMS VHF, bridge-to-bridge, interferometry, direction finding			**15. NUMBER OF PAGES** 85
			16. PRICE CODE
17. SECURITY CLASSIFICATION OF REPORT Unclassified	**18. SECURITY CLASSIFICATION OF THIS PAGE** Unclassified	**19. SECURITY CLASSIFICATION OF ABSTRACT** Unclassified	**20. LIMITATION OF ABSTRACT** UU

NSN 7540-01-280-5500 Standard Form 298 (Rev. 2-89)
Prescribed by ANSI Std. 239-18

THIS PAGE INTENTIONALLY LEFT BLANK

Approved for public release; distribution is unlimited

INEXPENSIVE SOLUTIONS FOR DIRECTION FINDING OF BRIDGE-TO-BRIDGE RADIO TRANSMITTERS USING PHASE DIFFERENCE IN RECEIVED SIGNAL

Eric Paul Boernke
Lieutenant, United States Navy
B.S., United States Naval Academy, 2003

Submitted in partial fulfillment of the
requirements for the degree of

MASTER OF SCIENCE IN ELECTRICAL ENGINEERING

from the

NAVAL POSTGRADUATE SCHOOL
March 2009

Author: Eric Paul Boernke

Approved by: Frank E. Kragh
Thesis Advisor

Herschel H. Loomis, Jr.
Co-Advisor

Jeffrey B. Knorr
Chairman, Department of Electrical and Computer Engineering

THIS PAGE INTENTIONALLY LEFT BLANK

ABSTRACT

This thesis presents two methods for determining the bearing of a source generating a Very High Frequency (VHF) bridge-to-bridge radio transmission. Using principles of interferometry, one can utilize the calculated phase difference between the received signal at multiple antennas to determine the Angle of Arrival of the detected transmission. This translation of phase difference to Angle of Arrival is accomplished using equations based on signal properties and geometry. The theoretical method is shown proving the relationship between Angle of Arrival and phase difference, as well as how a single platform could accomplish this detection and calculation. Theoretical simulation was accomplished using various simulation tools including Mathwork's Simulink and Tonne Software's Elsie. Methods are then provided to detect the phase difference using both a series of analog mixers and filters as well as digitally, using software radio. Analog filters were built and tested to determine the relationship between phase difference and voltage output. Software programs were written for a Software-Defined Radio implementing digital filtering to verify the analog performance. Results and accuracy are shown based on initial testing.

THIS PAGE INTENTIONALLY LEFT BLANK

TABLE OF CONTENTS

THIS PAGE INTENTIONALLY LEFT BLANK

LIST OF FIGURES

THIS PAGE INTENTIONALLY LEFT BLANK

LIST OF TABLES

THIS PAGE INTENTIONALLY LEFT BLANK

EXECUTIVE SUMMARY

Situational awareness at sea is essential for the navigational safety and security of the United States' military vessels. The Navy routinely operates in high-traffic areas with large volumes of shipping, including straits, canals, and other waters close to shore. As the volume and shipping density increases, increasing coordination is required between vessels, both military and civilian, to ensure the safety of all. Much of that coordination is conducted over Very High Frequency (VHF) bridge-to-bridge Radio. This maritime system operates between 156 and 162 MHz with 50 kHz of bandwidth assigned to each channel. It is by using this system that ships hail one another and conduct conversations regarding maneuvering intentions and other safety issues. In areas of high traffic, there can be nearly continuous chatter on this system. Unless one is participating in the conversation, it can be very difficult for many ships to determine who is talking at the time and from where the conversation is originating. Even if not involved in the conversation, it could be vital that a United States naval vessel be aware of the maneuvering intentions of the ships around it, for its own safety and security. Most conversations on bridge-to-bridge radio either originate or are conducted on channel 16, which is located at 156.8 MHz. This channel is reserved for international distress, safety, and calling; most ships and coastal stations maintain a listening watch on this channel. This thesis focuses on this frequency, but the methods conducted could be used to achieve similar results on any channel in the bridge-to-bridge VHF frequency range.

Previous research into the area of direction finding has focused on using techniques such as Time Difference of Arrival (TDOA) and Frequency Difference of Arrival (FDOA). The drawback to these systems is that they either require multiple transmitting platforms or multiple receiving platforms. While United States naval vessels traditionally operate in groups, there are situations where they are transiting alone. In these situations, many ships would benefit from a direction-finding system that can obtain a solution with only one transmitting platform and one receiving platform. Furthermore, the research focuses on simpler, low cost solutions to this problem. This can be accomplished using multiple antennas on a single platform and measuring the

phase differences seen at each antenna. This technique will enable a single ship to determine the bearing of vessels transmitting on their bridge-to-bridge radios.

This thesis examined two possible solutions to this problem, an analog solution and a digital solution. The analog solution was explored using traditional Radio Frequency (RF) components: resistors, capacitors, and inductors. These were utilized along with traditional RF mixers to achieve a solution. Rapid improvements in technology, however, have led to many functions traditionally found in hardware to now be implemented in software. In many cases, software-based solutions can be preferable as they can be cheaper, more flexible, and achieve higher performance. It is these advantages that led to the development of the software radio. There are exceptions, of course; in this research, the hardware-based solution proved cheaper than the software solution.

Ideally, the software-defined radio receiver would be nothing more than an analog-to-digital converter (ADC) attached to an antenna. The ADC would digitize the analog signal, and then computer software would process all the results and perform any needed applications or calculations. However, due to the desired frequencies of the signals of interest, oftentimes a downconverter is required to bring the signal to a lower intermediate frequency or baseband before digital sampling can occur. The device utilized in this thesis to do both downconversion and analog to digital conversion is the Universal Software Radio Peripheral (USRP).

The objectives of this research were to:

- Review existing methods of using phase difference of a received signal as an inexpensive means of determining a line of bearing to the transmitter.
- Model the overall system using computer simulations.
- Construct circuit boards based on design results
- Test the analog system to verify performance
- Use software to achieve a digital solution

- Test the digital system to verify performance
- Compare analog and digital results to determine the preferred solution to the research problem

These objectives of this research were met with different levels of success for the analog and digital systems.

The feasibility of using phase difference as a means of obtaining the Angle of Arrival was verified by mathematical calculations, which showed a direct link between the phase difference of one signal received at two antennas and the Angle of Arrival from which the signal originated. The system was then implemented in Simulink® to determine the feasibility of implementing the mathematical solution. While the Simulink simulation proved highly successful, its direct implementation in hardware proved infeasible due to the high frequency signal (156.8 MHz) and the relatively narrow bandwidth (50 kHz). The design of the system was then altered to take into account the hardware limitations, and the system was re-simulated. The system proved successful, which led to the design of the hardware.

The design required a series of analog filters which were then modeled in Elsie to ensure feasibility with industry's currently manufactured RF component ratings. Once the design was complete and verified, circuit boards were constructed using commercial off-the-shelf RF components and surface-mounted soldering to circuit boards. These boards were tested successfully and demonstrated a clear relationship between the voltage output of the system and the phase difference.

Concurrently, a digital system was designed using the GNU radio and Octave software projects. In this system, all filtering was performed digitally and the output was sent to a personal computer where the results were displayed. As in the analog case, there was a clear relationship seen between the output and the phase difference.

To further make this system compatible for shipboard use, the following additional research is required:

- Test using wireless signals to determine system effectiveness.
- Implement Low Noise Amplifiers and Automatic Gain Control to account for real-world environmental conditions.
- Test in a shipboard environment to determine maritime suitability.

With free and open sea lines of communication remaining vital to the United States economic and vital military interests, further research is recommended, for this research shows excellent potential for an inexpensive solution to the bridge-to-bridge radio direction-finding problem, thereby increasing situational awareness for United States warships.

ACKNOWLEDGMENTS

I would like to thank the many faculty and staff of the Naval Postgraduate School for their immense help and assistance throughout the entire thesis process, especially Professor Frank Kragh, Professor Herschel Loomis, Bob Broadston, and Donna Miller. Your time, knowledge, expertise, and advice are greatly appreciated.

My sincerest thanks must go to my beautiful wife Brianne. Without her unending love, support, and understanding, this would not have been possible.

THIS PAGE INTENTIONALLY LEFT BLANK

I. INTRODUCTION

A. BACKGROUND

Situational awareness at sea is essential for the navigational safety and security of the United States' military vessels. The Navy routinely operates in high-traffic areas with large volumes of shipping, including straits, canals, and other waters close to shore. As the volume and shipping density increases, more coordination is required between vessels, both military and civilian, to ensure the safety of all. Much of that coordination is conducted over Very High Frequency (VHF) bridge-to-bridge Radio. This maritime system operates between 156 and 162 MHz with 50 kHz of bandwidth assigned to each channel [1]. It is by using this system that ships hail one another and conduct conversations regarding maneuvering intentions and other safety issues. In areas of high traffic, there can be nearly continuous chatter on this system. Unless one is participating in the conversation, it can be very difficult for many ships to determine who is talking at the time and from where the conversation is originating. Even if not involved in the conversation, it could be vital that a United States naval vessel be aware of the maneuvering intentions of the ships around it, for its own safety and security. Most conversations on bridge-to-bridge radio either originate or are conducted on channel 16, which is located at 156.8 MHz. This channel is reserved for international distress, safety, and calling; most ships and coastal stations maintain a listening watch on this channel. This thesis focused on this frequency, but the methods conducted could be used to achieve similar results and other channel in the bridge-to-bridge VHF frequency range.

Navigational safety was not the only relevant concern prior to commencing work on this thesis. A proposed solution for using bridge-to-bridge radio as a direction-finding tool could help provide added safety and security to many ships and their crews. A major motivator to this research was an incident that occurred in the Strait of Hormuz on January 6, 2008. On that day, three United States Navy ships were completing a routine transit through the strait when they were aggressively approached by five Iranian small boats. It was over bridge-to-bridge radio that the small boats communicated their threats against the U.S. ships, telling them that they would soon explode [2]. In this scenario, the

warships did have visual contact with the aggressive small boats, but a situation could be envisioned where threats are received from ships outside visual range or are received from an area with high traffic density. In littoral environments, the signals could also be originating from shore-based installations. The only way to tell who is transmitting is to trust vessels to identify themselves truthfully or to have a system to map signals to their sources. In these situations, a system utilizing bridge-to-bridge radio as a direction-finding tool would be of great benefit to many warships. This system would give a line of bearing in the direction from which the threats are being transmitted. In a situation where no other ships are in sight, it would give the warship extra time to get ready for the threat and a direction as to where it should focus its efforts. In a heavy shipping area, it would help the warship distinguish potential threats from the mass of surrounding traffic. In both cases, the safety and security of the warship would be enhanced, making it better prepared for any potential threat.

This thesis examined two possible solutions to this problem, an analog solution and a digital solution. The analog solution was explored using traditional Radio Frequency (RF) components: resistors, capacitors, and inductors. These components were utilized to create filters that, combined with traditional RF mixers, achieved a solution. Rapid improvements in technology, however, have led to many functions traditionally found in hardware to now be implemented in software. In many cases, software-based solutions are preferable as they can be cheaper, more flexible, and achieve higher performance. It is these advantages that led to development of the software radio [3]. The primary disadvantage to a software solution is the relatively recent development of this field. It may take time to develop a software solution to a problem that has traditionally been solved with hardware. The final solution with software may be preferable when compared with hardware, but the time to develop that solution may prove problematic if a strict timeline exists.

Ideally, the software-defined radio receiver would be nothing more than an analog-to-digital converter (ADC) attached to an antenna. The ADC would digitize the analog signal, and then computer software would process all the results and perform any needed applications or calculations. However, due to the desired frequencies of the

signals of interest, oftentimes a downconverter is required to bring the signal to a lower intermediate frequency or baseband before digital sampling can occur. The device utilized in this thesis to do both the downconversion and analog to digital conversion is the Universal Software Radio Peripheral (USRP) [3].

This research explored whether a single platform would be able to perform direction finding via an inexpensive and simple solution and whether the phase difference that occurs between one signal arriving at two antennas could be utilized for that purpose. This thesis sought to answer both those questions through both theory and experimentation.

B. OBJECTIVES AND APPROACH

The objectives of this research were to:

- Review existing methods of using phase difference of a received signal as a means of determining a line of bearing to the transmitter.
- Model the overall system using Simulink
- Model the needed analog filters using Elsie, Tonne Software's electrical filter design and analysis program
- Construct circuit boards based on design results
- Test the analog system to verify performance
- Use GNU Radio and Octave programs to achieve digital solution
- Test the digital system to verify performance
- Compare analog and digital results to determine the preferred solution to the research problem

The feasibility of using phase difference as a means of obtaining the Angle of Arrival was verified by mathematical calculations, which showed a direct link between the phase difference of one signal received at two points and the Angle of Arrival from which the signal originated. The system was then implemented in Simulink to determine

the feasibility of implementing the mathematical solution. While the Simulink simulation proved highly successful, its implementation in hardware proved infeasible due to the high frequency signal (156.8 MHz) and the relatively narrow bandwidth (50 kHz). The design of the system was then altered to take into account the hardware limitations, and the system was re-simulated. The system proved successful, which led to the design of the hardware.

The design required a series of analog filters which were then modeled in Elsie to ensure feasibility with industry's currently manufactured RF component ratings. Once the design was complete and verified, circuit boards were constructed using commercial off-the-shelf RF components and surface-mounted soldering to circuit boards. These boards were tested successfully and demonstrated a clear relationship between the voltage output of the system and the phase difference.

Concurrently, a digital system was designed using GNU radio and Octave programming [3, 4. In this system, all filtering was performed digitally and the output was sent to a personal computer where the results were displayed. As in the analog case, there was a clear relationship seen between the output and the phase difference.

C. RELATED WORK

The issue of direction finding as a research field is not new. Much research has been previously conducted exploring various techniques and methods to find a working solution. In 2006, Derek Elsanesser published a paper detailing his Discrete Probability Density Method for geolocation [5]. His solution was well suited for geolocation utilizing large numbers of measurements from different locations, for it effectively fused the data from multiple sensors. In this research, however, the 'sensor' utilized is the already-existing shipboard bridge-to-bridge radio receiver. Additionally, the measurements are not received from multiple platforms, but rather from two antennas on one platform.

Other research into direction finding focused on using information other than phase to achieve the geolocation solution. Three popular techniques are: Angle of Arrival (AOA), Frequency Difference of Arrival (FDOA), and Time Difference of

Arrival (TDOA). Geolocation using Angle of Arrival requires lines of bearing taken at multiple locations at a single instance of time. Triangulation is then used to generate a solution from the individual lines of bearing. Time Difference of Arrival utilizes cross-correlation or some other delay-estimation technique from two spatially separated, moving receivers to generate a geolocation solution. Frequency Difference of Arrival is similar to TDOA but utilizes the difference in Doppler shifts between two spatially separated receivers to achieve the geolocation solution. There has been significant research into different ways of fusing these methods together to achieve a more precise geolocation solution [6, 7, 8].

The use of phase difference as a source of direction-finding is known as interferometry. It has been used as a common technique in electronic warfare direction-finding theory [9]. This thesis was concerned with the scenario of a single vessel steaming alone and needing to determine the location of a transmitter over bridge-to-bridge radio. This solution would also prove to be very cost-effective as the new equipment required would be minimal for it uses equipment already installed on the vast majority of marine vessels.

Recently, many students at the Naval Postgraduate School have written theses on software-defined radio (SDR), but only a few have explored SDR as a method for direction finding. One thesis that proved particularly useful was Ian Larsen's 2007 thesis which used SDR for direction finding [10]. However, his research focused on correlation-based Time Difference of Arrival to achieve a solution. As stated previously, this thesis focused on both hardware and software solutions with the signal's phase difference as the basis of the solution.

D. THESIS ORGANIZATION

The body of this thesis is divided into three chapters to indicate the level of technical detail. Chapter II details the analysis and derivation of the phase difference-based solution to the direction-finding problem. Chapter III focuses on the hardware solution to this problem, including the design, experimentation, and results of the analog hardware solution. Chapter IV details the software solution to this problem, including

general information on Software-Defined Radio, the Universal Software Radio Peripheral (USRP), and the results obtained. Chapter V summarizes results, makes conclusions, and recommends future work.

II. ANALYSIS

This chapter describes the theory of interferometry and its mathematical basis. Mathematical relationships are then developed which show the basis for calculating the phase difference, which will be used to calculate the Angle of Arrival.

A. INTERFEROMETRY

As stated earlier, the basis for this research's direction-finding solution is the phase difference that exists due to the arrival of a signal at two different points. This method of using the difference in phase to determine the Angle of Arrival of an incoming signal is known as interferometry. Figure 1 shows a graphical representation of the interferometry solution.

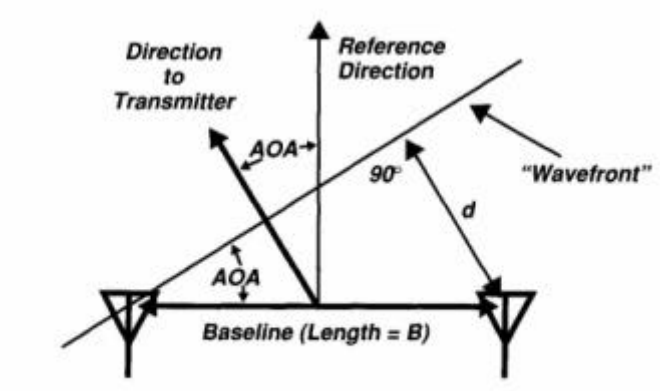

Figure 1. Graphical Representation of Interferometry. (From [9]).

The direction to the transmitter is labeled as *AOA*. The distance between the two receiving antennas (baseline) is defined as B. *d* is the distance between an antenna and the incoming wavefront. The mathematical relationships between these variables is [9]

$$AOA = \arcsin\left(d/B\right). \tag{2.1}$$

To calculate *d*, one uses the formula [9]

$$d = \frac{\phi c}{360 f} \tag{2.2}$$

where *c* is defined as the speed of light $\left(\approx 3.00 \times 10^{8} \, \mathrm{m/s}\right)$, *f* is the frequency of the transmitted signal, and ϕ is the phase difference (in degrees) of the received signal arriving at the two antennas.

Using these formulas, one can see that the more phase change required to change the Angle of Arrival, the more accurate the direction-finding system will be. In other words, greatest accuracy will be achieved for angles near perpendicular to baseline and worst at angles near ends of the baseline. Additionally, accuracy will improve as the length of the baseline, relative to the transmitted signal's wavelength, is increased. This is true up to a baseline equal to one-half of the wavelength. After that point, the phase difference changes more than 360° as the AOA moves from $+90^{\circ}$ to -90° [9].

Rearranging Equation (2.1) yields

$$d = B \sin\left(AOA\right). \tag{2.3}$$

Combining (2.2) and (2.3) yields

$$\begin{aligned} B \sin\left(AOA\right) &= \frac{\phi c}{360 f_c} \\ AOA &= \arcsin\left(\frac{\phi c}{360 \cdot B \cdot f_c}\right). \end{aligned} \tag{2.4}$$

The above equation proves that the Angle of Arrival of the transmitted bridge-to-bridge signal can be calculated if one knows the phase difference encountered by the two separate antennas. The value of c is a known constant, and the variables, *B* and *f*, are known, for the user will know the distance between the receiving antennas, as well as the frequency of the signal one is capturing. Once the phase difference is known, one must use that information to calculate the Angle of Arrival. Angle ambiguities arise in that calculation, which will be detailed further in Chapter III.B.1.c.

B. CALCULATING PHASE DIFFERENCE

As stated earlier and seen in (2.4), the key to determining the Angle of Arrival is knowing the phase difference of the incoming signal. This can be accomplished by multiplying the incoming signal with its phase-shifted version and then performing low-pass filtering. With the use of Automatic Gain Control and Low Noise Amplifiers, one is able to set the amplitude of the signals seen by the system. A cosine wave represents the carrier wave of the bridge-to-bridge signal. Assuming unit amplitude, the incoming signal is given by

$$\cos\left(2\pi f_c t\right) \tag{2.5}$$

and the phase shifted signal is given by

$$\cos\left(2\pi f_c t + \phi\right). \tag{2.6}$$

Multiplying the two signals together, with output *a(t)*, yields

$$\begin{aligned} a(t) &= \left[\cos\left(2\pi f_c t + \phi\right)\right]\left[\cos\left(2\pi f_c t\right)\right] \\ a(t) &= \frac{1}{2}\cos\phi + \frac{1}{2}\cos\left(4\pi f_c t + \phi\right) \end{aligned} \tag{2.7}$$

Applying a Low Pass Filter to remove the high-frequency components yields the following relationship:

$$LPF\left\{a(t)\right\} = \frac{1}{2}\cos\phi \tag{2.8}$$

This research focused on incoming signals from a bridge-to-bridge radio system. Because a ship is a relatively slow-moving vessel, the phase difference will not be changing rapidly. Due to this fact, the output value of (2.8) is essentially a DC value. Thus, that value can be applied to (2.4) to calculate the Angle of Arrival. As stated earlier, angle ambiguities arise when ϕ is used to calculate Angle of Arrival, which will be detailed in Chapter III.B.1.c.

C. CALCULATING PHASE DIFFERENCE USING LOCAL OSCILLATOR

In the hardware solution, which will be detailed further in Chapter III, it proved necessary to introduce a local oscillator to calculate the phase difference at the two receiving antennas. The system with the local oscillator addition is shown in Figure 2.

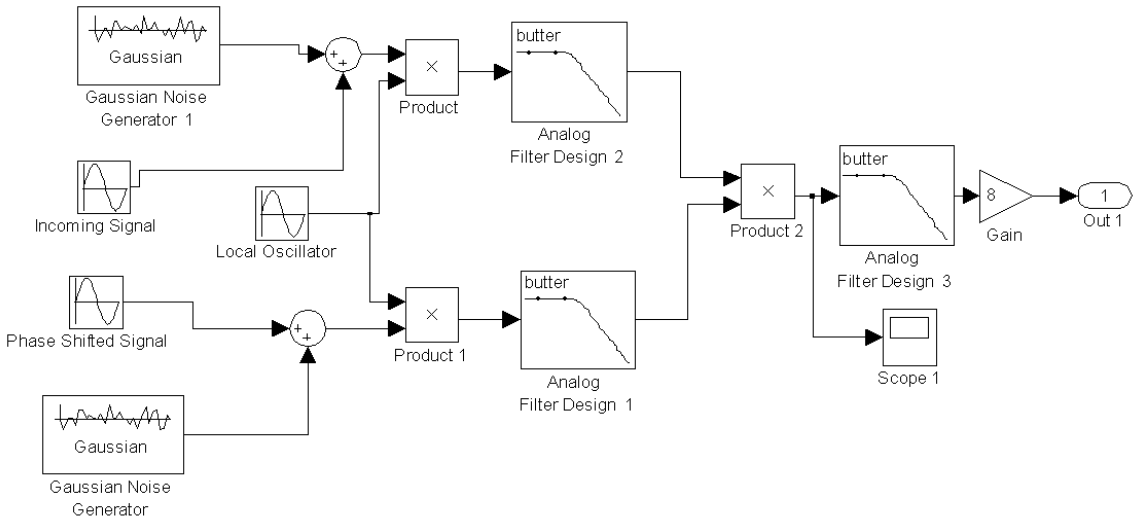

Figure 2. Direction-finding System with Addition of Local Oscillator.

This oscillator is used to downconvert the signals from the bridge-to-bridge frequency (between 156-162 MHz) to an intermediate frequency that is better suited for the filters in the system. As seen above in Section B, mixing the desired signal with its phase-shifted version enabled acquisition of the phase information. However, questions remained whether adding a local oscillator would still enable the phase information to be acquired. The following system of equations shows that the phase information is still able to be obtained, albeit with a more complicated system. For computational purposes, one assumes that the incoming signal has been adjusted to unit amplitude, and the local oscillator output has unit amplitude as well. The mixing of the incoming signal with the output of the local oscillator, yields output

$$\begin{aligned} S_n &= \cos\left(2\pi f_c t\right)\cos\left(2\pi\left(f_c + z\right)t + \theta\right) \\ &= 0.5\cos\left(2\pi\left(2f_c + z\right)t + \theta\right) + 0.5\cos\left(2\pi z t + \theta\right) \end{aligned} \tag{2.9}$$

where the desired intermediate frequency is given by *z* and the phase difference between the incoming signal and the local oscillator is given by θ. Applying a low pass filter to remove the high-frequency $(4\pi f_c t)$ components yields the following relationship:

$$LP\{S_n\} = \frac{1}{2}\cos(2\pi zt + \theta) \tag{2.10}$$

A similar method can be applied to the phase-shifted version of the desired signal. The following equations show the mixing of the incoming phase-shifted signal with the output of the local oscillator, yielding output S_p. As before, the desired intermediate frequency is given by *z.*

$$\begin{aligned} S_p &= \cos(2\pi f_c t + \phi)\cos(2\pi(f_c + z)t + \theta) \\ S_p &= \frac{1}{2}\cos\left[2\pi(2f_c + z)t + \phi + \theta\right] + \frac{1}{2}\cos(2\pi zt + \theta - \phi) \end{aligned} \tag{2.11}$$

Applying a low pass filter to remove the high-frequency $(4\pi f_c t)$ components yields the following relationship:

$$LP\{S_p\} = \frac{1}{2}\cos(2\pi zt + \theta - \phi) \tag{2.12}$$

Thus from (2.12) one can see that the phase information is still preserved despite the addition of the local oscillator. Yet from (2.10) and (2.12), one can still see the local oscillator frequency remains in S_n and S_p. Thus, the signals need to be mixed again in order to achieve a function that solely depends on phase difference. As seen before, it is the phase information that is needed to obtain the Angle of Arrival. Using similar notation as before, *a(t)* is the mixed signal.

$$\begin{aligned} a(t) &= LP\{S_n\}\,LP\{S_p\} \\ a(t) &= \frac{1}{2}\cos(2\pi zt + \theta) \times \frac{1}{2}\cos(2\pi zt + \theta - \phi) \\ a(t) &= \frac{1}{8}\cos\phi + \frac{1}{8}\cos(4\pi zt + 2\theta - \phi) \end{aligned} \tag{2.13}$$

Applying a low pass filter to remove the double intermediate frequency $(4\pi zt)$ components yields

$$LP\{a(t)\} = \frac{1}{8}\cos\phi \tag{2.14}$$

This derivation shows that the solution is independent of the phase of the local oscillator. Like the result seen in (2.8), angle ambiguities arise when ϕ is used to calculate Angle of Arrival which will be detailed in III.B.1.c. Comparing (2.8) and (2.14), one can see that both methods yield results that are based solely on the phase difference. It was based on the result of (2.14) that design could proceed using the local oscillator method. Because a ship is a relatively slow-moving vessel, the phase difference will not be changing rapidly. Due to this fact, the output value of (2.14) is essentially a DC value. Thus, the phase can be applied to (2.4) to calculate the Angle of Arrival.

Thus, one can see that is mathematically feasible to calculate the phase difference that occurs when one signal arrives at two different points. This information can then be used to calculate the Angle of Arrival from where the signal originated. Based on this foundation, work was undertaken to implement these findings in analog hardware, and this work is covered in the next chapter.

III. HARDWARE SOLUTION

This chapter details the method of building the direction-finding system using analog hardware components. Simulation tools and techniques are described. Finally, the process of building the circuit boards is described, and results are presented.

A. INTRODUCTION

As shown in Chapter II, it proved mathematically feasible to determine the Angle of Arrival from the phase difference between two signals received at two separate antennas. Thus, an effective proof of concept was realized. The system was viable and capable of solving the problem investigated by this research. However, the equations and mathematical methods utilized were completely idealized. Critical assumptions that were made included:

- Received signals were completely free from noise.
- All mixing was achieved with zero loss.
- All filters were completely ideal.

The next step in the design process was to electronically simulate the entire system using established computer software before any hardware or components were purchased. This would ensure that the funds would not be wasted building a system that would not even perform in a computer environment. The simulation was performed on a system-wide level using Mathwork's Simulink® [11] program, as well as on a system component level using Tonne Software's Elsie™ [12].

B. SIMULATION TECHNIQUES

This section describes the system-wide and lower level simulation techniques utilized in the hardware solution.

1. System-wide Simulation Using Simulink

This section describes the system-wide simulation conducted without using a local oscillator. It then details the reason for the addition of a local oscillator to the system.

a. System-wide Simulation without Local Oscillator

From Mathwork's corporate website, Simulink is described as "an environment for multidomain simulation and Model-Based Design for dynamic and embedded systems[11]." This software program was a logical next step in the design process, for a system-wide simulation was needed to test the proof of concept derived earlier. Simulink has a wide array of block libraries that lets the user design, simulate, and test both simple and complex systems. There are many user-customizable settings for the overall system, as well as for individual blocks within the system. These settings are able to be quickly changed, greatly enhancing the user's ability to test the system. The system can be put through a wide range of testing conditions all from a personal computer. Simulink is also integrated with Mathwork's flagship program, MATLAB®. Thus, the user is able to take advantage of MATLAB's computational power as well as its programming capabilities. One also has the ability to run MATLAB scripts to develop companion algorithms to enhance the simulation, as well as to define signal, parameter, and test data [11]. It was these factors, along with Simulink's prevalence in the engineering community, which made it a logical choice to model the desired system.

Figure 3 shows the Simulink Model of the direction-finding system described in Chapter II.

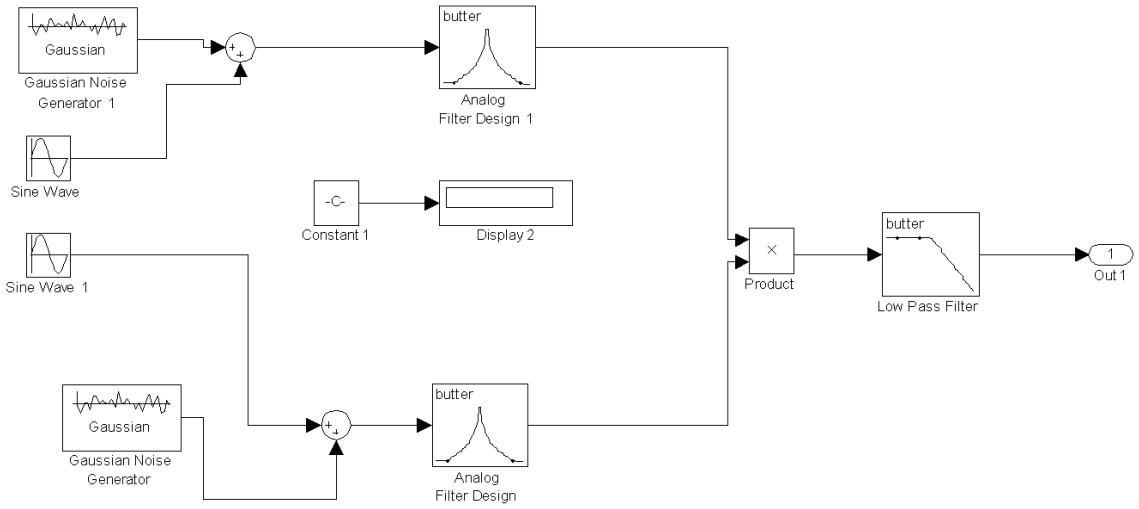

Figure 3. Initial Implementation of Direction-finding Model.

The top Sine Wave block simulates the non-phase-shifted signal seen at one antenna. It has unit amplitude and a frequency of 156.8 MHz (bridge-to-bridge Radio Channel 16). White Gaussian noise is added to that sinusoid in order to simulate the noise added to a signal in a real-world environment. The combined signal is then passed through a Butterworth bandpass filter. The bandpass filter is needed so that the user can select which channel on the bridge-to-bridge frequency spectrum is of interest and to eliminate out of band signals, noise, and interference. The characteristics of the bridge-to-bridge frequency spectrum necessitate a very narrow bandpass filter. The bandwidth of each channel is only 50 kHz [1], which is much less than 1 percent of the channel frequency. This narrow passband and the close location of adjacent channels required a steep roll-off on either side of the passband. Thus, to only capture and analyze one channel, a high order bandpass filter is required. For the simulation seen in Figure 3, a Butterworth Filter with order 5 was used to analyze the system.

When analyzing the above simulation, the passband was originally set strictly for Channel 16, with 156.8 MHz as the center frequency and 156.775 MHz and 156.825 MHz as the 3 dB cut-off frequencies. This reflects the 50 kHz of bandwidth assigned to each bridge-to-bridge channel. With such a narrow passband, however, the system was not able to achieve an acceptable Angle of Arrival. The solution also did not

improve as the order of the filter was increased. Regardless of the filter order, the system was not able to determine the Angle of Arrival of the incoming signal. This was due to the fact that trying to achieve such a narrow passband resulted in unacceptable attenuation of the signal.

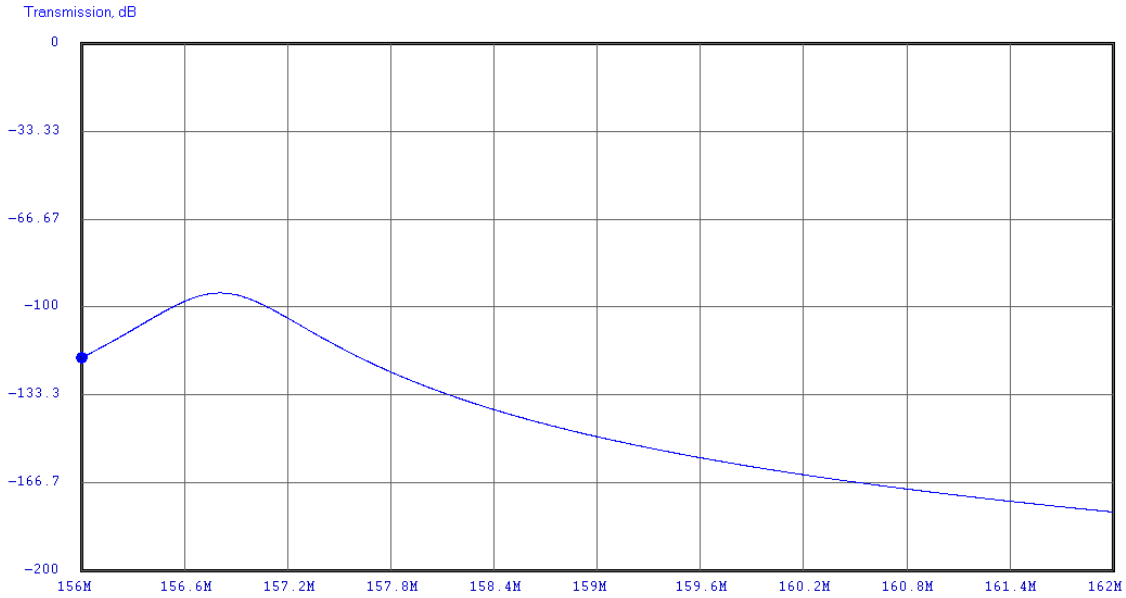

Figure 4. Bandpass Filter Centered at 156.8 MHz and 50 kHz Bandwidth.

Figure 4 shows that with such tight parameters, there is almost 100 dB of attenuation at the center frequency. Due to this, the filter parameters had to be adjusted in order for the system to determine the Angle of Arrival.

In order for the system to achieve an acceptable Angle of Arrival, the passband of the bandpass filter needed to be widened. The bandpass filter was then expanded to include the range of frequencies used by bridge-to-bridge radio, from 156.05 MHz to 162.025 MHz [1]. The plot of the bandpass filter with these parameters is shown in Figure 5.

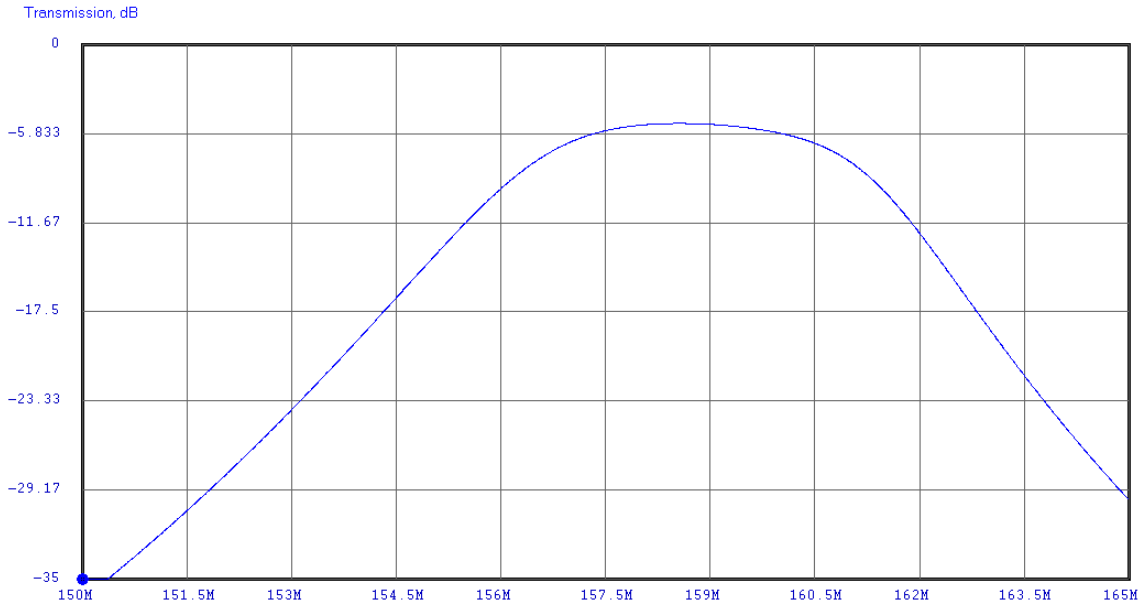

Figure 5. Bandpass Filter Centered at 159 MHz and 6 MHz Bandwidth.

By widening the passband, one can see that the passband attenuation has improved to approximately -5 dB. However, this has eliminated the ability of the filter to focus on one specific channel. If there is only one bridge-to-bridge channel being transmitted at a specific instant of time, then the system will be able to ascertain the Angle of Arrival of that signal. That scenario is unlikely, especially so in areas of high shipping density. In those situations, the more likely situation is that multiple ships will be having multiple conversations on a variety of bridge-to-bridge channels. When this happens, the system will not be able to track a specific signal, which is the overall purpose of the system. A bandpass filter, as shown in Figure 5, would be able to filter out other signals in the Very High Frequency band, such as FM radio broadcasts, air traffic control, and NOAA weather stations [1].

Because this research was intended to be used on United States naval vessels, which regularly operate in areas close to shore and in areas of high traffic, the above findings necessitated a re-design. A solution needed to be found that would enable the system to focus on one specific bridge-to-bridge channel. This would, in turn, eliminate all surrounding signals in the Radio Frequency spectrum. In addition to the

above requirements, the system would need to be able to be built and manufactured with off-the-shelf commercial components that are readily available and economical.

b. System-wide Simulation Utilizing Local Oscillator

As shown in Chapter II, Section C, it proved theoretically and mathematically possible to introduce a local oscillator to the system and still recover the desired phase information. The local oscillator would also address and solve the problems introduced in the previous section. Most importantly, utilizing a local oscillator in the system would eliminate the need for a bandpass filter. The system would become more complex with the addition of a signal generator and an additional stage of mixing. However, these additions would enable the system to be implemented using a combination of low-pass filters, which are significantly simpler to design, build, and implement.

Utilizing a local oscillator would also enable the vessel operating the direction-finding system to focus on a specific bridge-to-bridge channel. Using the variables defined in Chapter II, one is able to design the system and first stage filters about a specific intermediate frequency *z*. Knowing the intermediate frequency, one can set the local oscillator frequency to $\left(f_c + z \right)$. Thus, the system will only yield a coherent solution if the frequency of the incoming signal received at the antenna is equal to the local oscillator frequency, less the intermediate frequency. If they are equal, one is able to obtain the relationships defined in (2.10) and (2.12), shown again here for convenience.

$$LP\left\{S_n\right\} = \frac{1}{2}\cos\left(2\pi zt + \theta\right) \tag{3.1}$$

$$LP\left\{S_p\right\} = \frac{1}{2}\cos\left(2\pi zt + \theta - \phi\right) \tag{3.2}$$

Conversely, if the frequency of the incoming signal received at the antenna is not equal to the local oscillator frequency, less the intermediate frequency, the following relationship is seen:

$$S_n = \cos(2\pi f_n t)\cos(2\pi(f_c + z)t)$$

$$S_n = \frac{1}{2}\cos[2\pi(f_c + z - f_n)t] + \frac{1}{2}\cos[2\pi(f_c + z + f_n)t] \quad (3.3)$$

where f_c is the frequency of the desired bridge-to-bridge radio channel and f_n is the frequency of another signal, possibly another signal in the bridge-to-bridge frequency spectrum. When the incoming signal frequency matches the local oscillator frequency, less the intermediate frequency, a low pass filter is able to filter out high frequency components to achieve the relationships shown in (3.1) and (3.2). Applying a low pass filter will filter out the cosine and sine terms with a frequencies of $(f_n + f_c + z)$. However, the cosine and sine terms with frequencies of $(f_c + z - f_n)$ may not necessarily be eliminated. This is especially true if referring to two signals within the overall bridge-to-bridge frequency spectrum. Signals in that spectrum are separated by no more than 2 MHz, so the $(f_c + z - f_n)$ terms would not be eliminated by the low pass filter, with a 3 dB cut-off frequency. If those terms are not eliminated, one is not able to achieve relationships based solely on phase. One will never know f_n due to the fact that it is an interfering signal, thus one will not be able to receive a coherent solution if the frequency of the signal at the antenna is not equal to the local oscillator frequency, less the intermediate frequency. One way to ensure that the $(f_c + z - f_n)$ terms would be eliminated would be to put a tight bandpass filter on the front end of the system, with the passband set precisely at the desired f_c. Another option would be to use a bandpass filter with 50 kHz bandwidth centered at the intermediate frequency. This filter would be used in lieu of the low-pass filters in the system. As stated earlier, such narrow bandpass filters proved infeasible to implement. These areas would benefit greatly from future research.

The Simulink model shown in Figure 3 needed to be adapted to reflect the addition of the local oscillator to the system. The updated and final Simulink model of the direction-finding system is shown in Figure 6.

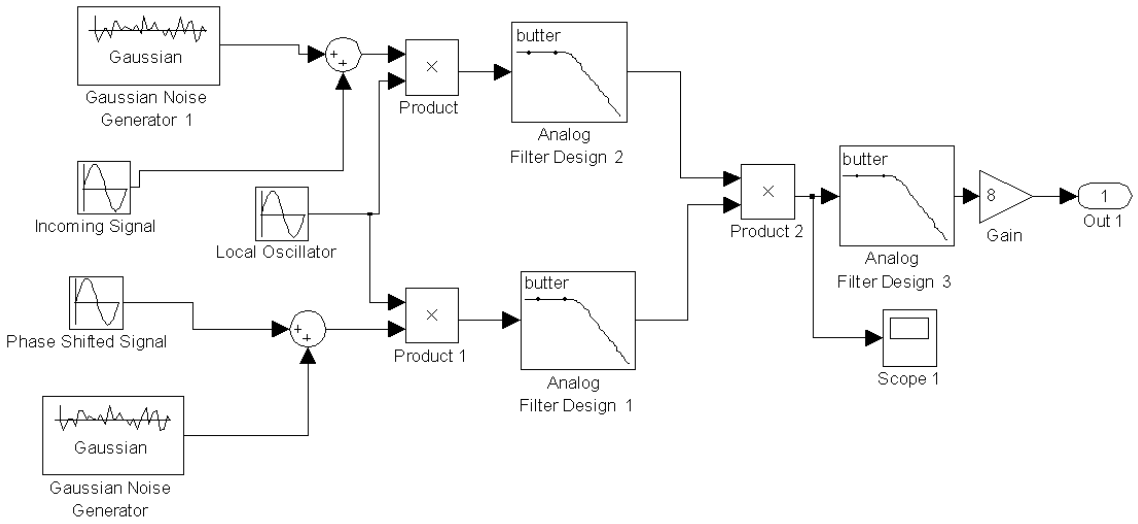

Figure 6. Simulink Model with Local Oscillator.

As before, sinusoids represent the incoming bridge-to-bridge transmission. For the simulation, they were given unit amplitude and set at 156.8 MHz, representing Channel 16 on bridge-to-bridge radio. In this final model, a third Sine Wave block has been added to represent the local oscillator. The oscillator frequency was set to 167.5 MHz (156.8 + 10.7 MHz). The intermediate frequency (IF) was chosen to be 10.7 MHz, due to the prevalence of this frequency as an IF in FM radio circuits [13]. Because this system is meant to be an economical solution to the direction-finding problem, using technical specifications that are common in the commercial world will lead to lower implementation and fielding costs of the system.

Additive white Gaussian noise is added to the incoming signal and its phase-shifted version in order to simulate the noise encountered in a real-world environment. These signals are then multiplied by the local oscillator output and passed through a low-pass filter. In Simulink, these filters were modeled by third-order Butterworth low-pass filters. The cut-off frequency was set to the IF frequency of 10.7 MHz, which will pass the information contained at that frequency and below but eliminate the information at the original carrier and higher frequencies. After this first round of filtering, the system has the relationships seen in (2.10) and (2.12). These signals, S_n and S_p, are then mixed together and passed through another low-pass filter.

This low pass filter is designed to filter out the intermediate frequency and leave an essentially DC signal that is a function of the phase difference. This low-pass filter was given a cut-off frequency of 1 kHz. This value was chosen as it is significantly lower than 10.7 MHz, but it also accounts for the fact that in a real-world environment the final signal will not be purely DC. After filtering, the signal is multiplied by a constant value of eight. After that multiplication, the signal is now equal to $\cos\phi$.

c. Requirement for Two Systems

After the signal passes through the constant gain of eight, calculations must be performed to transform the calculated phase into an Angle of Arrival. The first step in this process is taking the inverse cosine of the signal, which should yield the phase difference. However, the inverse cosine is only defined on the interval $[0,\pi]$. What this means is that for every value of $\cos\phi$, there are two possible phases on $[0,2\pi]$. Thus, the system must then be split into two to take into account the two possible phases.

The system for translating the phase difference into Angle of Arrival is shown in Figure 7.

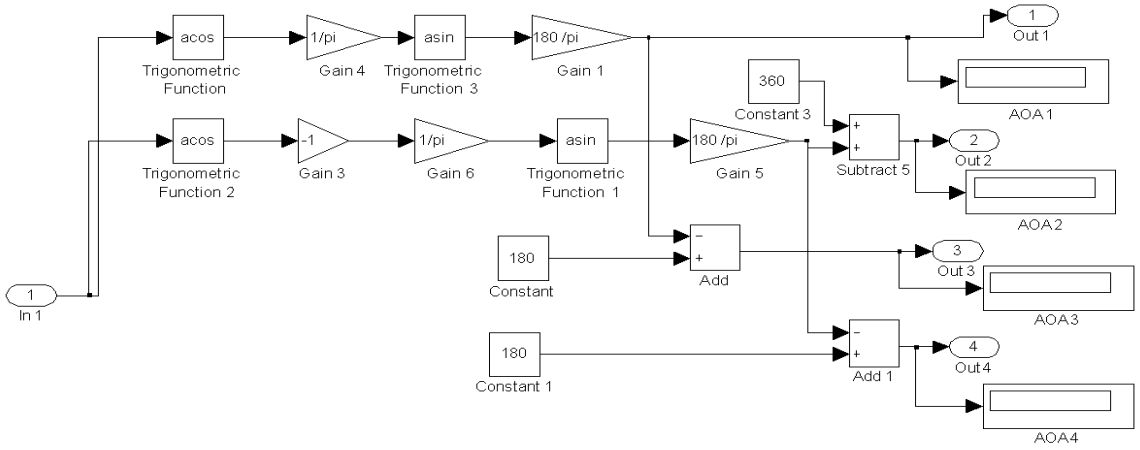

Figure 7. System for Translating Phase Difference to Angle of Arrival.

As seen above the system is split into two arms to account for the two possible solutions to the inverse cosine operation. The signal in the bottom arm is multiplied by negative one to account for angles between $[\pi, 2\pi]$. As shown previously in (2.4), the relationship between phase difference and AOA is:

$$AOA = \arcsin\left(\frac{\phi c}{360 \cdot B \cdot f_c}\right) \tag{3.4}$$

Converting (3.4) into radians gives the following relationship:

$$AOA = \arcsin\left(\frac{c\phi}{2\pi f_c B}\right) \tag{3.5}$$

When designing the Simulink model, it was assumed that the baseline distance between the two receiving antennas was equal to $\left(\lambda/2\right)$, where λ is the wavelength of the desired bridge-to-bridge signal. For bridge-to-bridge radio channel 16, $\lambda/2$ is approximately 0.96 meters. As stated in Ch. II, accuracy will improve as the length of the baseline, relative to the transmitted signal's wavelength, is increased. This is true up to a baseline equal to one-half wavelength. After that point, the phase difference changes more than 360° as the AOA moves from $+90^\circ$ to -90° [9]. Thus, assuming $b = \left(\lambda/2\right)$, (3.5) simplifies to:

$$AOA = \arcsin\left(\frac{\phi}{\pi}\right) \tag{3.6}$$

After performing the inverse cosine operation, one has two possible solutions for ϕ, one solution between $[0, \pi]$ and the other between $[\pi, 2\pi]$. One must multiply both values of ϕ by $\left(1/\pi\right)$ to achieve the quantity $\left(\phi/\pi\right)$. Because

$$\sin\left(AOA\right) = \left(\frac{\phi}{\pi}\right) \tag{3.7}$$

the final step to obtaining the Angle of Arrival is to take the inverse sine of the signals in both arms. However, like the inverse cosine, the inverse sine is only defined for a limited range of angles. The inverse sine is defined over the range $\left[-\frac{\pi}{2},\frac{\pi}{2}\right]$. Because of this property, the two arms must then be split into four branches to take into account the range of angles $\left[\frac{\pi}{2},\frac{3\pi}{2}\right]$. This will account for all possible Angles of Arrival. The mathematical operations to achieve all four possible Angles of Arrival are detailed below. For the first branch, after the arcsine operation is performed, the signal is multiplied by $\left(180/\pi\right)$ to convert the signal from radians to degrees. By performing this operation, the system is able to calculate Angles of Arrival between $\left[0^{\circ},90^{\circ}\right]$. The relationship between the output of the first branch and the third branch is:

$$\text{Third Branch Output} = \left(180 - \text{First Branch Output}\right) \tag{3.8}$$

This operation allows the system to calculate Angles of Arrival between $\left[90^{\circ},180^{\circ}\right]$. The second arm (in which the output of the inverse cosine was multiplied by negative one) is multiplied by $\left(180/\pi\right)$ to convert the signal to degrees. That signal is then added to 360 in order to obtain a positive Angle of Arrival; this produces the output of the second branch. By this operation, the system is able to calculate Angles of Arrival between $\left[270^{\circ},360^{\circ}\right]$. Finally, the relationship between the second and fourth branch is:

$$\text{Fourth Branch Output} = \left(180 \text{ - Second Branch Output}\right). \tag{3.9}$$

This operation takes into account Angles of Arrival between $\left[180^{\circ},270^{\circ}\right]$. While these operations add complexity to the system, they are necessary to ensure the system is able to calculate Angles of Arrival from the entire range of possibilities $\left[0^{\circ},360^{\circ}\right]$.

The purpose of this research was to develop a system that could be utilized onboard United States vessels with minimal user input. It should be a system from which the user can receive a clear solution and then act accordingly. If the system gives four possible answers, the user is left to decipher which answer is correct. In an environment in which traffic is sparse, this would not be a difficult task. One could simply look in the four directions and see clearly which of the four angles had other vessels on axis. Process of elimination would lead the watchstander to know which ship was transmitting. However, if traffic is heavy, there could be ships at each of the four possible Angles of Arrival. The watchstander would have absolutely no way of knowing which ship was transmitting. The system would need to be changed in order to eliminate the ambiguity a watchstander would encounter.

The solution that was determined was to introduce another pair of antennas offset from the original pair at a pre-determined angle. Thus, each pair of antennas would produce four solutions, and the solutions from Antenna Set 1 would be compared to the set of solutions from Antenna Set 2. By comparing solutions, one could find commonalities and determine the true Angle of Arrival of the transmitted signal. Different offset angles were experimented with to determine if a single, coherent solution could be obtained. The first offset attempted was a 90-degree offset of the second pair of antennas. Unfortunately, this amount of offset was unable to narrow down the solution set and four possible solutions remained. After multiple attempts with a variety of angle offsets, the best solution was obtained with an offset of 45 degrees. While this solution was not able to determine one single solution, it did narrow the solution set down from four possible solutions to two possible solutions. Each of the two antenna systems was analyzed at every Angle of Arrival between 0 and 360 degrees. The solution sets were then compared at every angle and certain patterns were noticed. At certain angles, for example, Solution Set 1 from antenna system 1 matched Solution Set 1 from antenna system 2. At the same angles, Solution Set 4 from antenna system 1 matched Solution Set 4 from antenna system 2. What made this finding important was that these solution

sets only matched at certain Angles of Arrival. Thus, by using a system of logic statements one was able to narrow the solution set down from four to two. Table 1 shows the input to the logic statements used to narrow the solution set.

	Matching Solution Sets			
True Angle of Arrival (degrees)	**System 1**	**System 2**	**System 1**	**System 2**
0-45	AOA1	AOA1	AOA4	AOA4
45-90	AOA1	AOA3	AOA4	AOA2
90-135	AOA3	AOA3	AOA2	AOA2
135-180	AOA3	AOA4	AOA2	AOA1
180-225	AOA4	AOA4	AOA1	AOA1
225-270	AOA4	AOA2	AOA1	AOA3
270-315	AOA2	AOA2	AOA3	AOA3
315-360	AOA2	AOA1	AOA3	AOA4

Table 1. Method for Determining Solution Set.

For example, if Antenna System 1 returns the solutions in Table 2:

AOA1	30
AOA2	330
AOA3	150
AOA4	210

Table 2. Example Solution Set from Antenna System 1.

and Antenna System 2 returns the solutions in Table 3 (after subtracting out the angle offset of the second antenna system),

AOA1	30
AOA2	240
AOA3	60
AOA4	210

Table 3. Example Solution Set from Antenna System 2.

one can clearly see that AOA1 from Antenna System 1 matches AOA1 from Antenna System 2, and AOA4 from Antenna System 1 matches AOA4 from Antenna System 2. Additionally, no values from either Antenna System's AOA2 or AOA3 are the same, so the solution set is down to two possible values: 30 degrees or 210 degrees. While this solution is not ideal for the watchstander, it has given him or her only two possible

solutions, separated by 180 degrees. The inability of this system to yield one definitive solution is a weakness in the system. Further research is recommended in this area to ascertain methods or algorithms to solve this problem.

2. Lower-level Simulation Using Elsie

Simulink effectively confirmed that the mathematical derivations could be implemented on a system-wide level and generate a set of two possible solutions. However, Simulink is still a highly idealized system. The Product blocks in Simulink were replaced by analog mixers when the system was actually built. The multiplication in Simulink is ideal, but analog mixing results in some power loss. Likewise, a perfectly ideal filter does not exist in the real world. When building an analog filter, achieving the exact cut-off frequency one desires is an unrealistic expectation. The next step in the design process was performing a more realistic simulation of the filters contained within the system. The software program chosen to perform this filter simulation was Tonne Software's Elsie, an electrical filter design and analysis program.

From Tonne Software's website, Elsie is described as a "32-bit Windows electrical filter design software nicely written to help engineers design and analyze lumped-element filters in the audio through microwave range." It allows the user to design lowpass, highpass, bandpass, and bandstop filters in many families of filters, including Butterworth, Chebyshev, Cauer, Bessel, and Gaussian. The student version of Elsie, which was utilized during this research, allows the design of filters up to seventh order. The program contains significant plotting capability, allowing thorough analysis of the designed filter. Once the ideal filter is designed, the program allows the user to substitute real-world component values into the system to see how the real-world filter differs from the theoretical version. Elsie proved a very capable program for linking the theoretical design of the filter to the building of physical filters [12].

The first step in using Elsie was to design the first stage of low-pass filters. As noted in the previous section, this filter was to pass frequencies up to the intermediate frequency of 10.7 MHz. The first attempt at the first-stage filter was that of a third-order Butterworth low-pass filter. The frequency response is shown below in Figure 8.

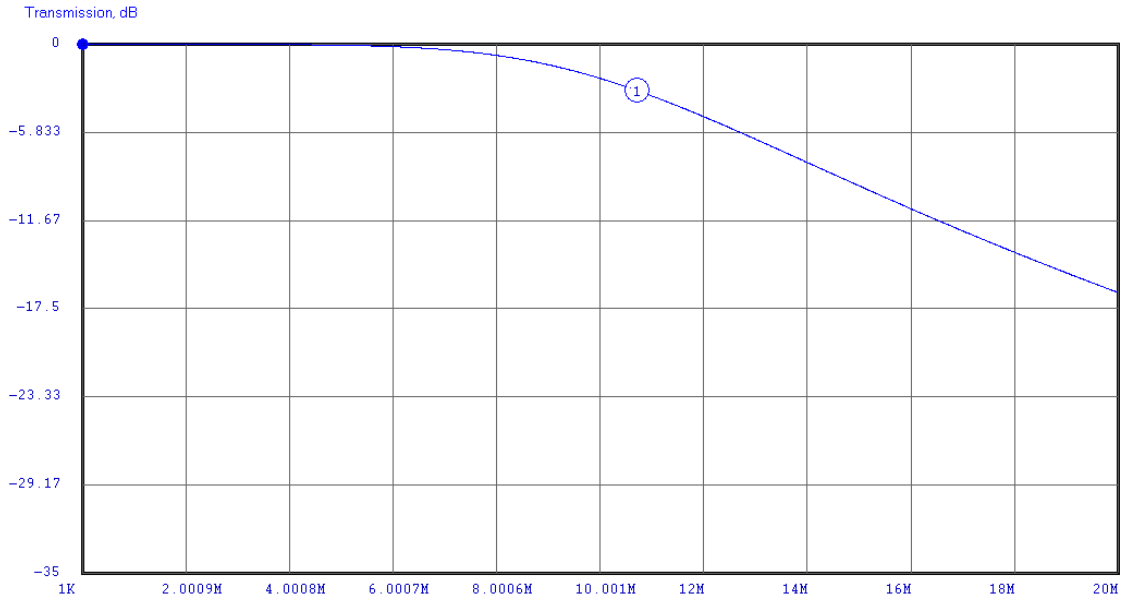

Figure 8. Ideal First Stage Low-Pass Filter.

The marker shown on Figure 8 is located at 10.7 MHz, with a transmission loss of 3 dB. This frequency response can be obtained with the third-order Butterworth filter shown in Figure 9:

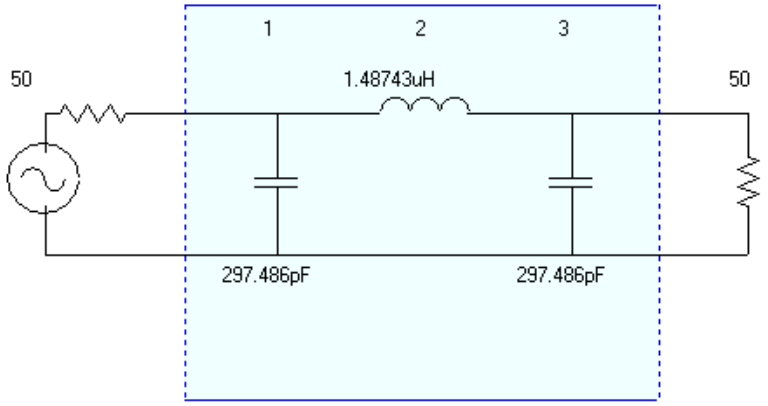

Figure 9. Ideal First Stage Low Pass Filter Schematic.

As one can clearly see in Figure 9, the capacitor and inductor values are purely theoretical. It is not possible to purchase such components accurate to the third and fourth decimal point commercially. Elsie was then used to convert these component values into components that could be purchased commercially. Figure 10 shows the schematic generated by Elsie after conversion to the commercial components.

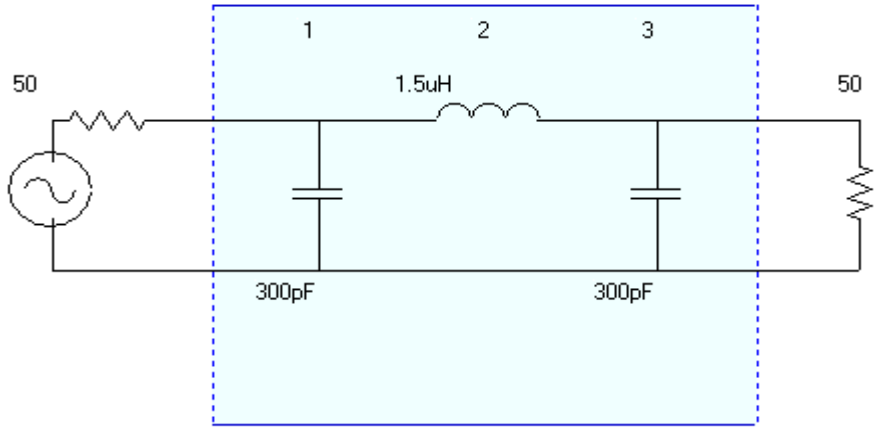

Figure 10. First-Stage Filter with Commercially Available Components.

Changing the component values naturally changed the frequency response of the filter. The changed frequency response is shown in Figure 11.

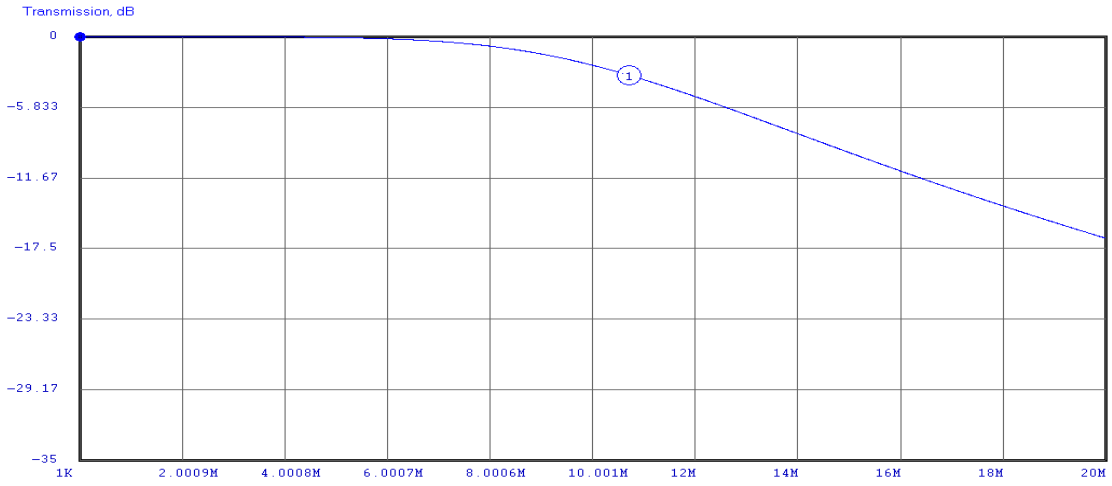

Figure 11. Frequency Response of First Stage Filter with Commercially Available Components.

By changing the component values, the transmission loss has increased from 3 dB to 3.17 dB at 10.7 MHz. This corresponds to a loss of 52 percent of the power after passing through just the first stage filter. Alternate filter designs were then explored with the possibility of lessening the power loss after the first-stage filter. After multiple iterations, the final design was reached with a cut-off frequency of 20 MHz. While this is greater than the intermediate frequency of 10.7 MHz, it is still significantly lower than the bridge-to-bridge frequency range, which begins at 156.05 MHz [1]. Staying significantly lower than the bridge-to-bridge frequency range ensures that the filter is capable of eliminating the information contained in the higher frequencies, as shown in (2.10) and (2.12). The frequency response of the final design of the first-stage filters is shown in Figure 12.

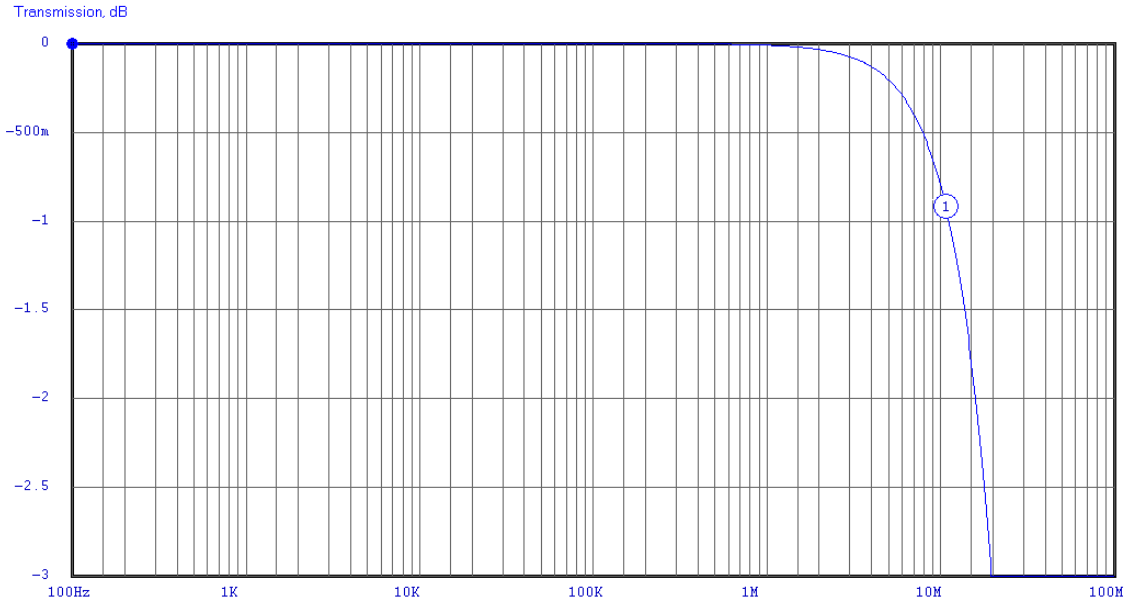

Figure 12. Frequency Response of Final Design of First-Stage Filters.

As in the previous figures, the marker is set at 10.7 MHz. This design led to significant improvement in the power loss, as there was just 0.91 dB of power loss at 10.7 MHz. This is equivalent to just 19 percent power loss, while still keeping the filter simple for construction purposes. The above performance was achieved with a third-order filter. Figure 13 shows the final filter schematic of the first-stage low-pass filters.

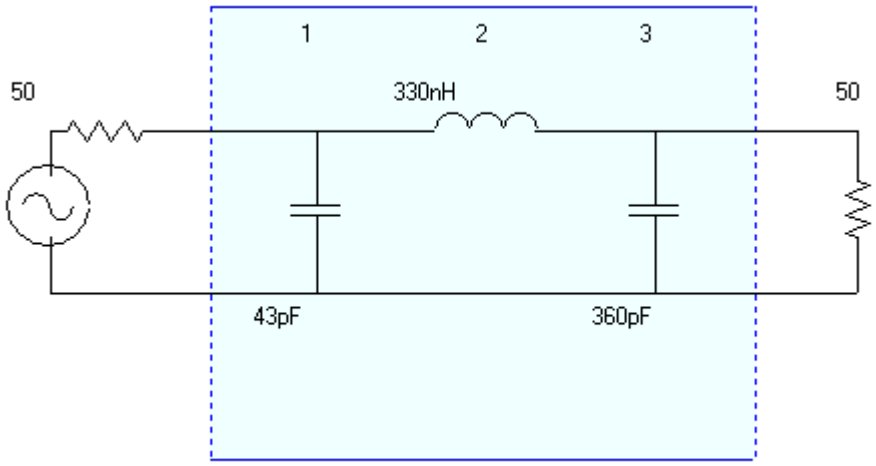

Figure 13. Circuit Schematic of Final Design of First-Stage Low-Pass Filters.

Finding a solution that could be accomplished with a third-order filter was important. Not only would the system be easier to implement, it would also be more cost-effective. Additional complexity would not only make building the system more difficult, but adding additional components would significantly escalate cost.

After the first-stage filters were designed, the next step was to design the second-stage filter. As noted in the previous section, this filter was to pass frequencies from DC up through 1 kHz. The first attempt at the second-stage filter was that of a third-order Butterworth low-pass filter. The frequency response is shown below in Figure 14.

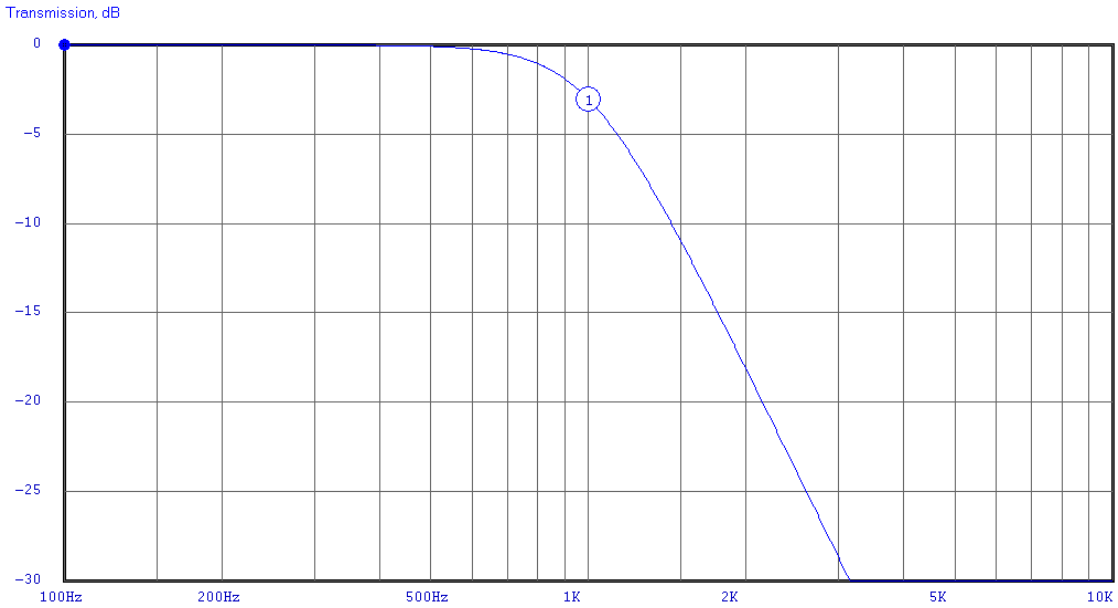

Figure 14. Ideal Second-Stage Low-Pass Filter.

The marker shown on Figure 14 is located at 1 kHz, with a transmission loss of 3.06 dB. The following filter design can be accomplished with a third-order Butterworth filter using the following schematic:

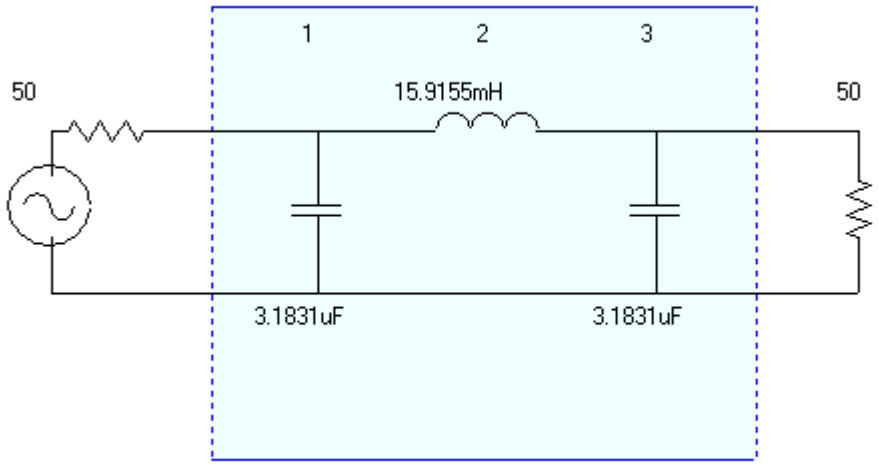

Figure 15. Ideal Second Stage Low Pass Filter Schematic.

As one can clearly see in Figure 15, the capacitor and inductor values are purely theoretical. It is not possible to purchase such components accurate to the fourth decimal point commercially. Elsie was then used to convert these component values into components that could be purchased commercially. Figure 16 shows the schematic generated by Elsie after conversion to the commercial components.

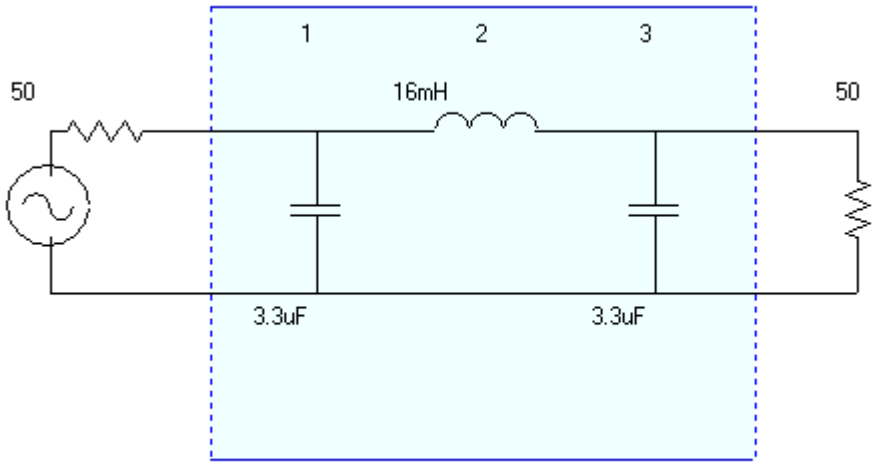

Figure 16. Second-Stage Filter with Commercially Available Components.

Changing the component values naturally changed the frequency response of the filter. The changed frequency response is shown in Figure 17.

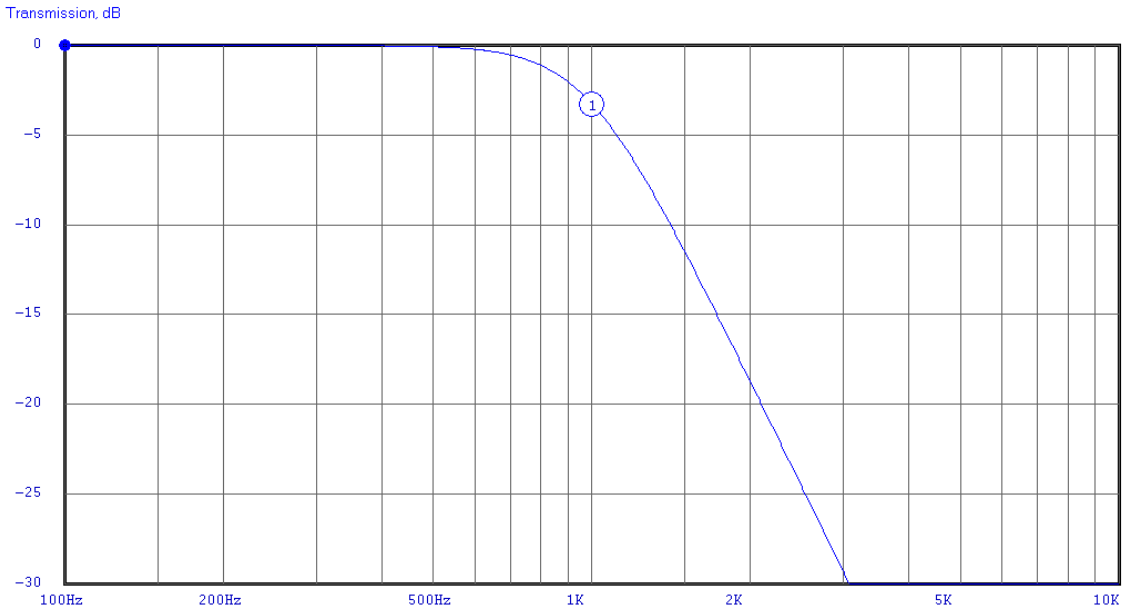

Figure 17. Frequency Response of Second Stage Filter with Commercial Parts.

By changing the component values, the transmission loss has increased from 3 dB to 3.27 dB. This corresponds to a loss of 53 percent of the power after passing through the second stage filter. Alternate filter designs were then explored with the possibility of lessening the power loss. After multiple iterations, the final design was reached with a cut-off frequency of 1 kHz. The frequency response of the final design of the first-stage filters is shown in Figure 18. The rise in the frequency response at approximately 6 kHz is due to the fact that a desired inductor was not commercially available and near-substitute component had to be used in the circuit construction.

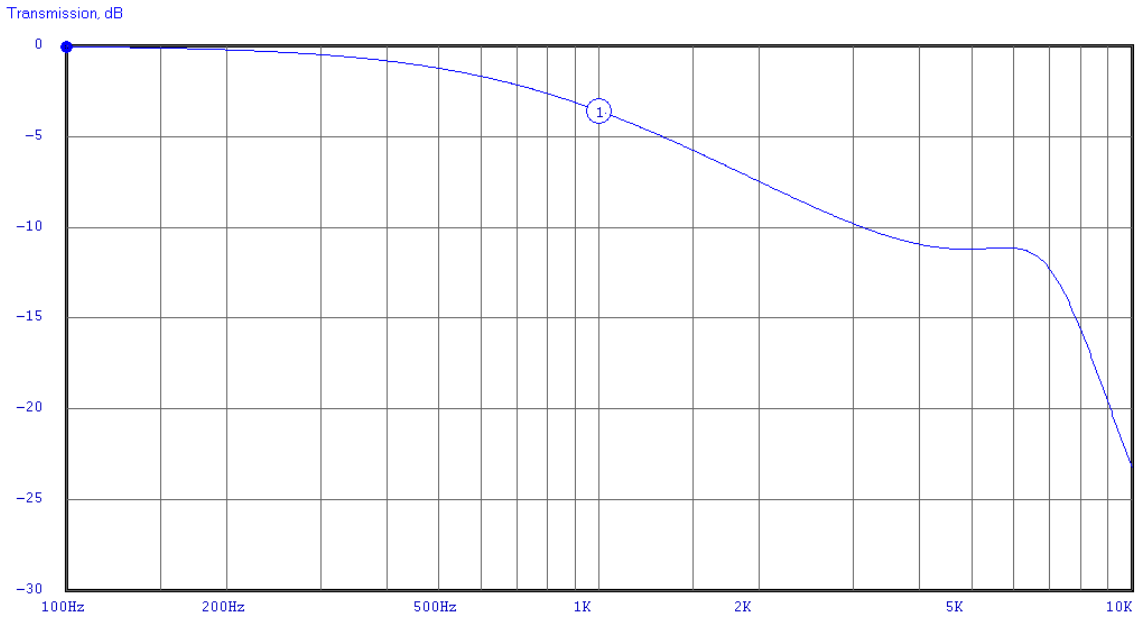

Figure 18. Frequency Response of Final Design of Second-Stage Filter.

Figure 19 shows the final filter schematic of the second-stage low-pass filter.

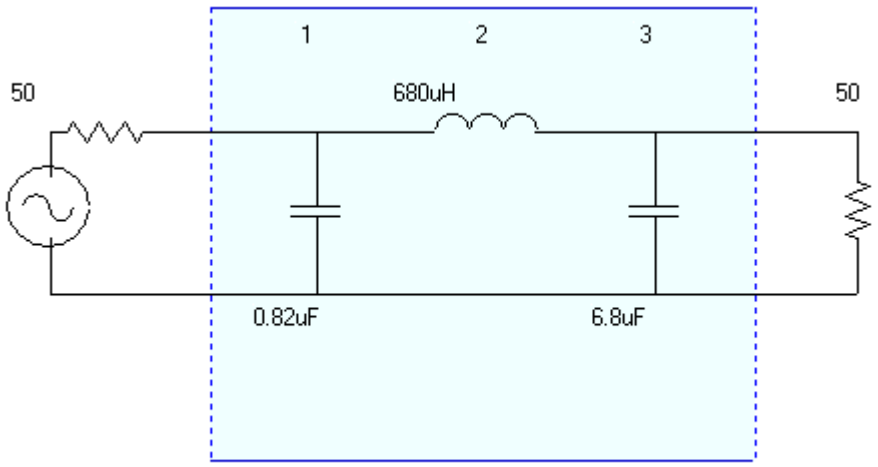

Figure 19. Circuit Schematic of Final Design of Second-Stage Low-Pass Filter.

As before, finding a solution that could be accomplished with a third-order filter was important. Not only would the system be easier to implement, it would also be more cost-effective. Additional complexity would not only make building the system more difficult, but adding additional components would significantly escalate cost.

Thus, the final system could be built utilizing only third order filters and commercial off-the-shelf components. By accomplishing these objectives, the next step of building the system would be even more manageable.

C. BUILDING THE HARDWARE SOLUTION

The next step in the design process was to build the circuits that would be required to implement the direction-finding system. Research was conducted into various electronics companies to ensure that the component values determined in the Elsie simulations were available for purchase. The companies, on the whole, carried the same components and in the same magnitudes. However, they differed in the availability of components, the lead time required for purchase, purchase minimums, and other factors. A major decision was whether to use through-hole components or surface-mounted components. Another consideration was the compatibility of the components with the expected test equipment. The capacitor values used in the first-stage filters are in the picofarad range, and the capacitors used in the second-stage filter are in the microfarad range. With such small values, the issue of stray capacitances was very important. Ultimately, surface-mounted components were chosen based on the fact that there would be much less stray capacitance than if through-hole components were used [14]. The connectors on the edge of each circuit board needed to be SMA (SubMiniature version A)-compatible, so they would be compatible with the test equipment. This requirement limited the number of connectors available for purchase. Ultimately, the most cost-effective connectors that fulfilled the design requirements were through-hole components. The capacitors, inductors, and resistors were able to be purchased as surface-mounted components.

Before the components could be purchased, the circuit boards on which they would be mounted needed to be designed and purchased. The company PCB123™, whose parent company is Sunstone Circuits®, was selected to manufacture these circuit boards [15]. PCB123 was ideal for a number of reasons. The design software for the circuit boards is available as a free internet download, with free customer and technical support. Once one is familiar with the software, the user is able to design the circuit boards with relative ease. The software comes with a comprehensive part library. This was advantageous because the boards could be designed with the exact components that were determined with the Elsie software. While the circuit boards required for this research needed only two electrical layers, PCB123 allows up to six layers. Another important feature was that after design was complete, the designs were sent electronically to PCB123 for manufacture. Not only would PCB123 manufacture the boards, the company would perform electrical checks afterwards to determine that the boards were manufactured correctly [15]. In the design process, the user is able to specify the locations and layout of all components, the size of the components, and the electrical path connecting the filters. Figure 20 shows the PCB schematic that incorporates both first-stage low-pass filters and the second-stage low-pass filter.

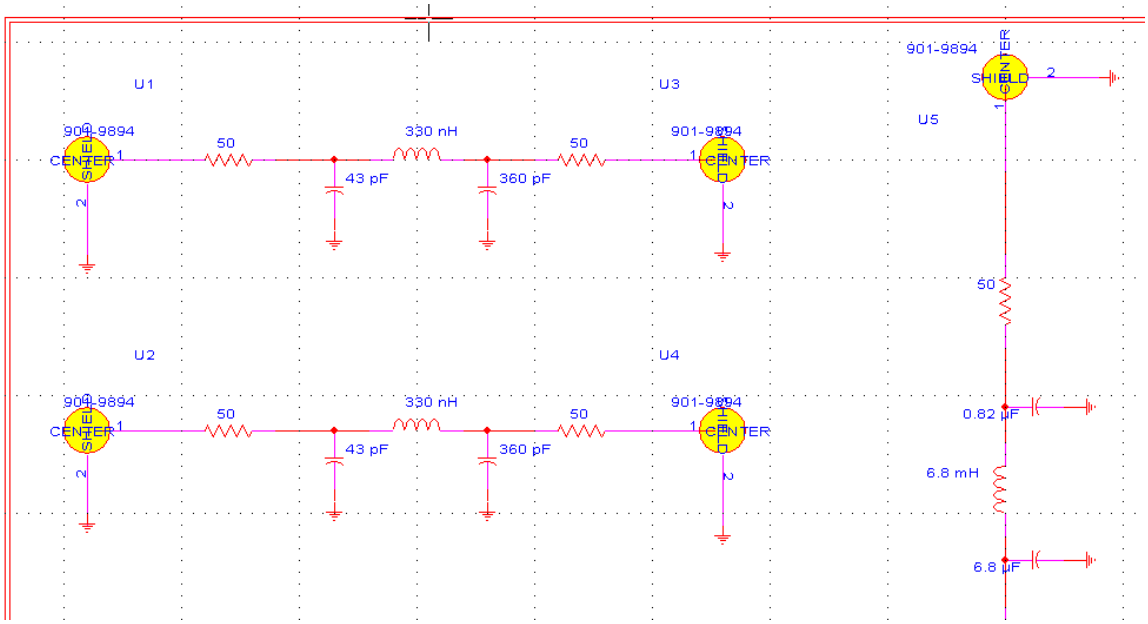

Figure 20. PCB123 Circuit Schematic.

Figure 20 shows the theoretical schematic of the direction-finding system on the circuit board. The PCB123 software converts the schematic to a "layout" model, where one can manipulate the physical location of components and electrical paths.

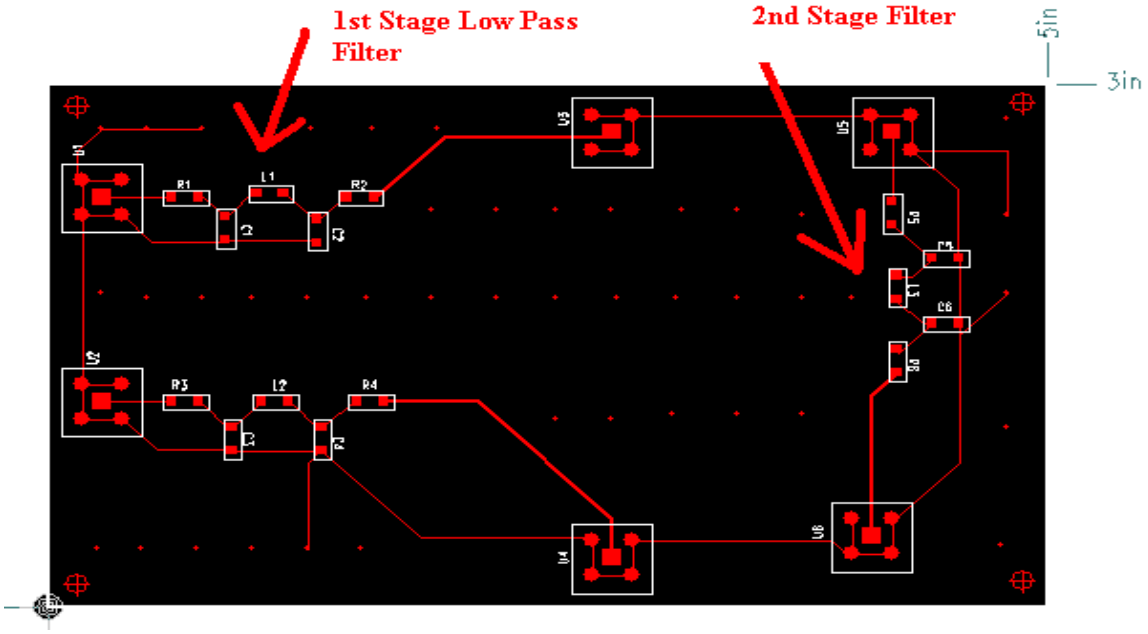

Figure 21. PCB123 Layout Model.

Figure 21 shows the layout model generated by PCB123. The boards chosen were three inch by five inch circuit boards. There were six SMA connectors per board, which are the large square objects seen near the borders in Figure 21.

The direction-finding system operates as follows: The incoming signal is multiplied by the local oscillator using an analog mixer, with the output entering the top first-stage low-pass filter from the left. The incoming phase-shifted signal is multiplied by the local oscillator using an analog mixer, with its output entering the bottom first-stage low-pass filter from the left. Both signals pass through their respective low-pass filters and then exit the board through the SMA connectors located in the middle of the circuit board. Those two signals are then multiplied together, by way of an analog mixer, and proceed to the second-stage low-pass filter. The output of the mixer re-enters the board through the top of the second-stage filter. The signal then passes through the second-stage low-pass filter and the DC voltage is read by a voltmeter for analysis.

D. HARDWARE SOLUTION RESULTS

After the circuit boards were received, the electrical components were ordered. The components were then soldered onto the circuit boards. Each antenna system requires one circuit board; thus, a minimum of two boards are required for the direction-finding system. Four were created, however, to increase the probability of having a functioning circuit board. After construction, the circuits were tested to ensure the filters met the design criteria. All four circuit boards were independently tested, and all four circuit boards yielded the same testing results. As stated in previous sections, the first-stage filters were built with a cut-off frequency of 20 MHz. Thus, there should be 3 dB of attenuation at 20 MHz. When tested, however, the first-stage filters experienced 3 dB of loss at 15 MHz. This was not a problem, though, for the desired intermediate frequency of 10.7 MHz was still well within the passband. In essence, the filter was tighter than designed but would have no negative effect on overall system performance.

Like the first-stage filters, the second-stage filter on each circuit board was tested independently. Each of the second-stage filters on the four circuit boards all had nearly identical performance. The second-stage filters were designed to have 3 dB of attenuation at 1 kHz. Upon testing, it was determined that there was actually 4 dB of attenuation at 1 kHz. While this is worse performance than originally designed, the signal of interest is basically DC, so the decreased performance at 1 kHz did not have an overall negative effect on system performance.

After verifying each circuit board's individual performance, the overall system was constructed. A signal generator was set at 156.8 MHz to simulate bridge-to-bridge radio Channel 16. The signal was also frequency modulated, to more realistically simulate an actual bridge-to-bridge radio signal. Another signal generator was used to simulate the local oscillator; this generator was set to 167.5 MHz, which reflects the bridge-to-bridge frequency plus the intermediate frequency. The system was then set-up as described in the previous section. The signal from the first signal generator was split, with one output unchanged and the second output passing through a Sage Laboratories® Model 6708 phase shifter [16]. This phase shift represented the phase shift that would be caused by the bridge-to-bridge signal arriving at the second antenna. Both signals were

multiplied by the local oscillator signal with a Mini Circuits® ZLW-1 analog mixer [17]. The local oscillator used was a Hewlett Packard 8341B Synthesizer Sweeper [18]. Both signals were then passed through the first-stage filters on one circuit board. After the signals exited the first-stage filters, they were mixed together with an analog mixer and then passed through the second-stage filter. The output of the second stage filter was then sent to a voltmeter for analysis. A picture of the overall testing system is shown in Figure 22.

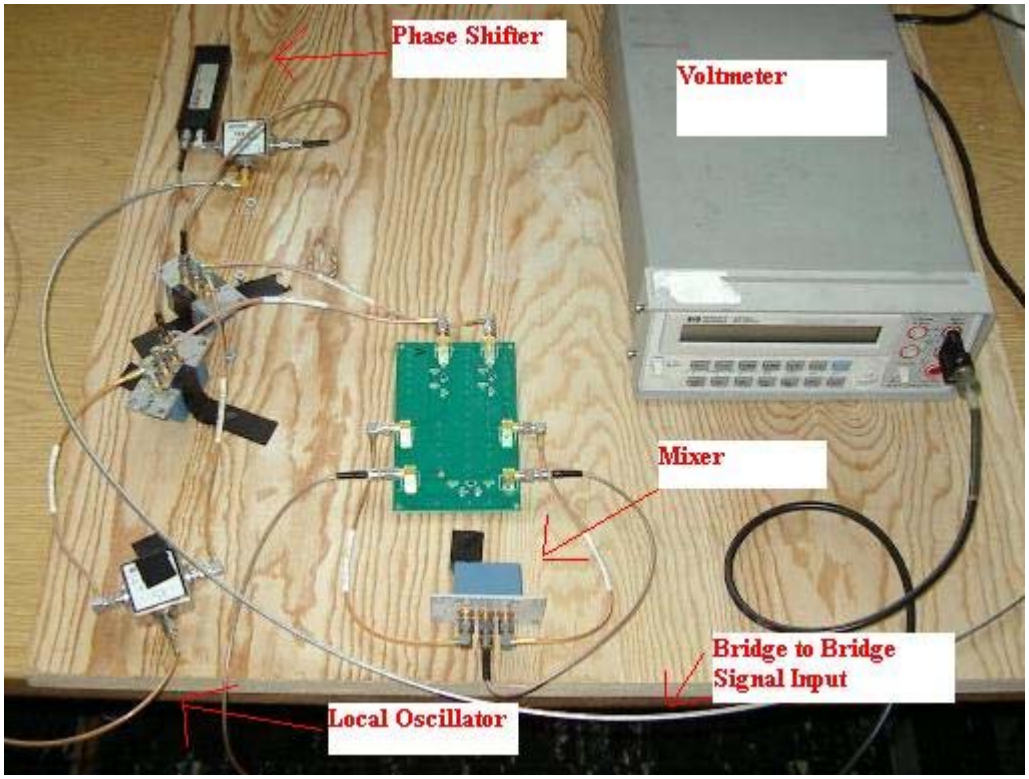

Figure 22. Hardware Solution Testing System.

The phase shifter used in the experiment varied the phase from zero to 180 degrees. The phase was changed by use of a knob on the phase shifter. As the knob was turned, a change in phase was introduced. The range of the phase shifter was used to determine how many degrees of phase shift were introduced by each turn of the knob. After the calibration of the phase shifter was verified, the experiment was then conducted

multiple times by comparing the voltage output of the system to the phase difference between the two original signals. Interestingly, throughout multiple trials, a clear linear relationship was observed. This was unlike what was seen in (2.14) and shown again below in (3.10)

$$LP\{a(t)\} = \frac{1}{8}\cos\phi \tag{3.10}$$

where a trigonometric relationship was derived. This was an unexpected benefit, as the periodic nature of trigonometric functions can yield solutions that are possible from more than one phase difference. The linear solution yields one unique solution per phase difference. Further research should be conducted to determine why the linear relationship was experienced, rather than the expected trigonometric relationship. The relationship is seen in the figure below, with the experimental data marked with circles and a best-fit line superimposed.

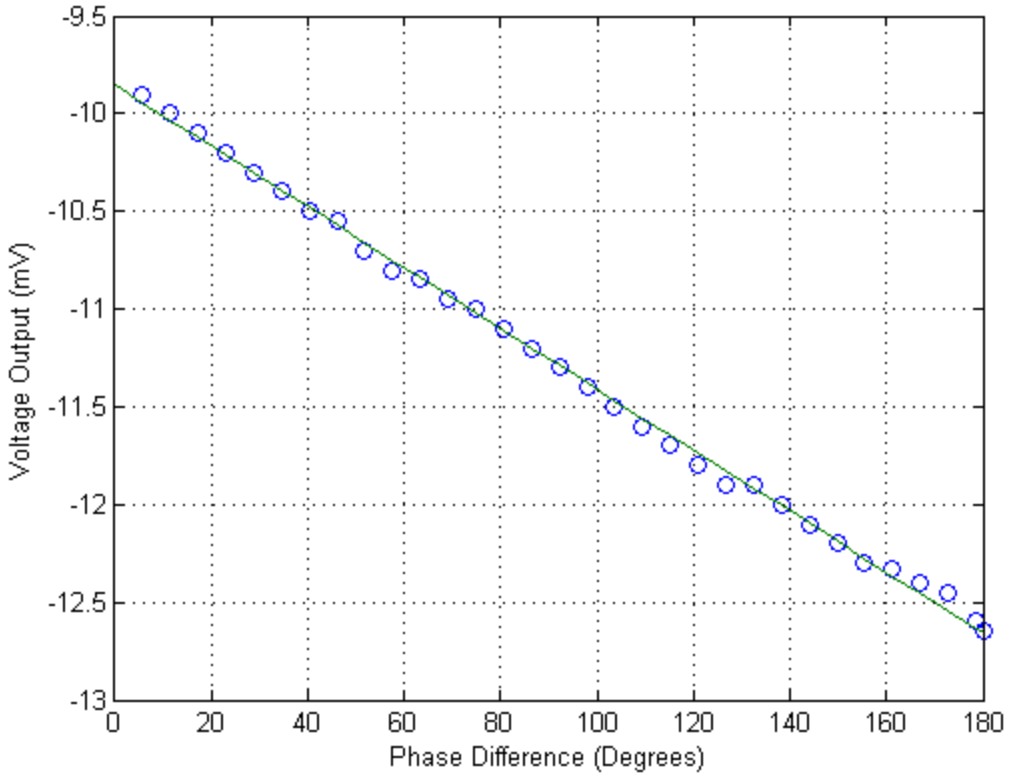

Figure 23. Relationship Between Phase Difference and Voltage Output.

One can clearly see the linear relationship between the phase difference and the voltage output. Using the equation for the best-fit line, the relationship between phase and voltage output is:

$$\text{phase}(\text{deg}) = \left(-62.5\frac{\text{deg}}{\text{mV}}\right) \times \text{voltage}(\text{mV}) - 615.625(\text{deg}) \tag{3.11}$$

This phase information is then entered into a MATLAB program that uses the equations derived in Chapter II to arrive at a solution set of possible Angle of Arrivals for Antenna System 1. Antenna System 2 is set up in the exact same way as Antenna System 1, but with a pre-determined angle offset. Comparing the solutions obtained from Antenna System 1 with the solutions obtained from Antenna System 2, one is able to determine the Angle of Arrival of the incoming signal.

Thus, the hardware solution has solved the problem for which the research was conducted. It has proved that a functioning system can be built with analog components that can accurately determine the Angle of Arrival of an incoming bridge-to-bridge signal if the phase difference between a single signal arriving at different locations is known. As the hardware solution was being explored, a software solution was being explored concurrently. Obtaining a solution in software would allow two different solutions to the direction-finding problem, which could be compared and analyzed to determine the best solution.

THIS PAGE INTENTIONALLY LEFT BLANK

IV. SOFTWARE SOLUTION

This chapter details the method of building the direction-finding system using software-defined radio. The software projects utilized are introduced and described, as well as the components used in the system. Finally, the process of setting up the experiment is described, and results are presented.

A. INTRODUCTION

As seen in Chapter II, it proved mathematically feasible to determine Angle of Arrival from the phase difference between two signals received at two separate antennas. The findings in Chapter III verified that a solution was possible using hardware analog components. Research was conducted concurrently to determine if a direction-finding solution could be obtained using software.

Rapid improvements in technology, have led to many functions traditionally found in hardware to now be implemented in software. Software-based solutions are often preferable as they can be cheaper, more flexible, and achieve higher performance. It is these advantages that led to the development of the software radio [19].

Ideally, the software-defined radio receiver would have an analog-to-digital converter (ADC) attached directly to an antenna. The ADC would digitize the analog signal, and then computer software would process all the results and perform any needed applications or calculations. However, due to the desired frequencies of the signals of interest, oftentimes a downconverter is required to bring the signal to a lower intermediate frequency or baseband before digital sampling can occur [10]. This device utilized in this thesis to do both downconversion and analog to digital conversion is the Universal Software Radio Peripheral (USRP) [20]. The first step in developing a digital solution to this problem was finding a software program that would be able to communicate with the USRP as well as be capable of implementing the direction-finding system.

B. GNU RADIO SOFTWARE APPLICATION

Two of the most popular open-source software projects for software-defined radio currently used are the GNU radio project and the Open Source SCA Implementation Embedded (OSSIE) project. The OSSIE project is a Virginia Tech University initiative to provide a Software-Defined Radio platform for the development of waveforms based on the Software Communications Architecture specifications under the Joint Tactical Radio System program. OSSIE, originally released in 2004, was developed primarily for research and education in software-defined radios. Additionally, the Naval Postgraduate School has worked with Virginia Tech in development of laboratory exercises. The OSSIE project is written in C++, and the focus on development is on the Linux operating system. However, there has been initial research performed on implementing OSSIE on other operating systems [21].

The other major Software-Defined Radio Project, and the project used in this research, is the GNU Radio Project. Like OSSIE, GNU Radio is a free, open-source Software-Defined Radio development platform. It contains low-level programs and different processing blocks to implement these radios. GNU Radio contains many libraries for modulation, error-correcting codes, and signal processing. GNU radio code is primarily written using the Python programming language, while the supplied, performance-critical signal-processing path is implemented in C++ using processor floating point extensions. The strength of GNU radio is that the waveforms transmitted and received by the radio are defined by software. This gives GNU radio and other software-defined radio systems their biggest advantage—reconfigurability. Multiple expensive radios no longer need to be purchased; a single generic radio can be purchased and then be programmed with signal processing software to perform the desired function. The same generic radio can then be reprogrammed if the desired functionality and purpose of the radio system changes [3].

For this research, the digital solution to the direction-finding problem was explored using GNU radio software. However, only half of the functionality of the software was utilized. This research did not require the use of the transmission capability of the radio. No waveforms needed to be developed, nor transmitted. The solution being

researched was a passive system, which required only the receive capability of the GNU radio software. The bridge-to-bridge radio signal of interest is an FM-modulated analog signal. Because of this property, an analog-to-digital converter (ADC) was needed that could convert the VHF signal to a digital signal that can be analyzed and processed by the GNU radio software. The Universal Software Radio Peripheral contained the ADC utilized in this research.

C. UNIVERSAL SOFTWARE RADIO PERIPHERAL

The Universal Software Radio Peripheral (USRP), developed by Matt Ettus, is a low-cost software radio device that connects to a personal computer using a USB 2.0 connection. It consists of a motherboard, which performs the analog-to-digital conversion of the baseband signal, and up to four daughterboards, which convert the original analog signals to the desired Intermediate Frequency (IF). The motherboard contains a Field Programmable Gate Array (FPGA) for high speed signal processing. The motherboard is shown in Figure 24 [20].

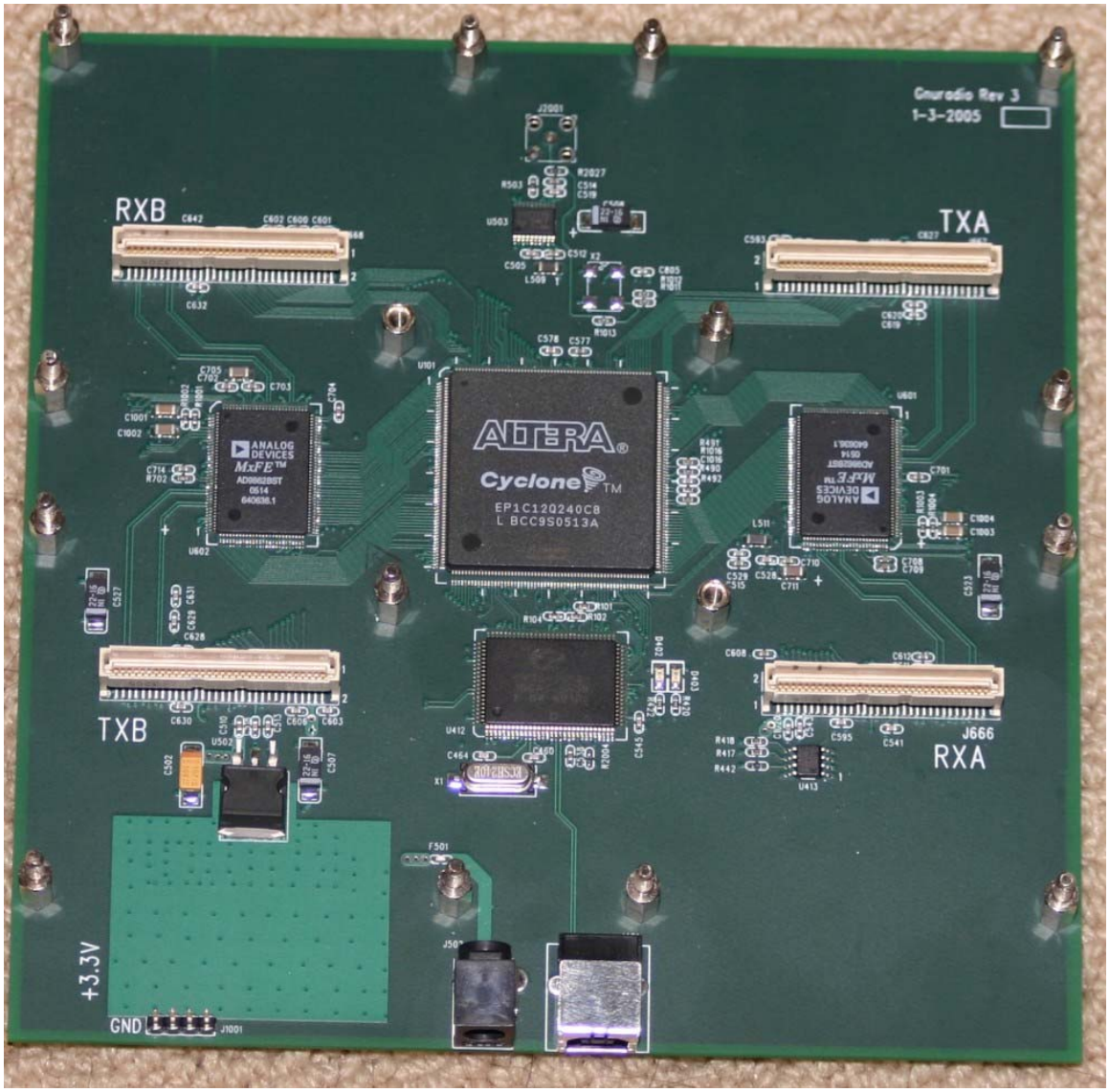

Figure 24. USRP Motherboard. (From [20]).

Each daughterboard can be used for a different frequency band, making the USRP an extremely flexible and useful device. Each motherboard can support up to two transmit daughterboards and up to two receive daughterboards. The basic receive daughterboard is shown in Figure 25 [20].

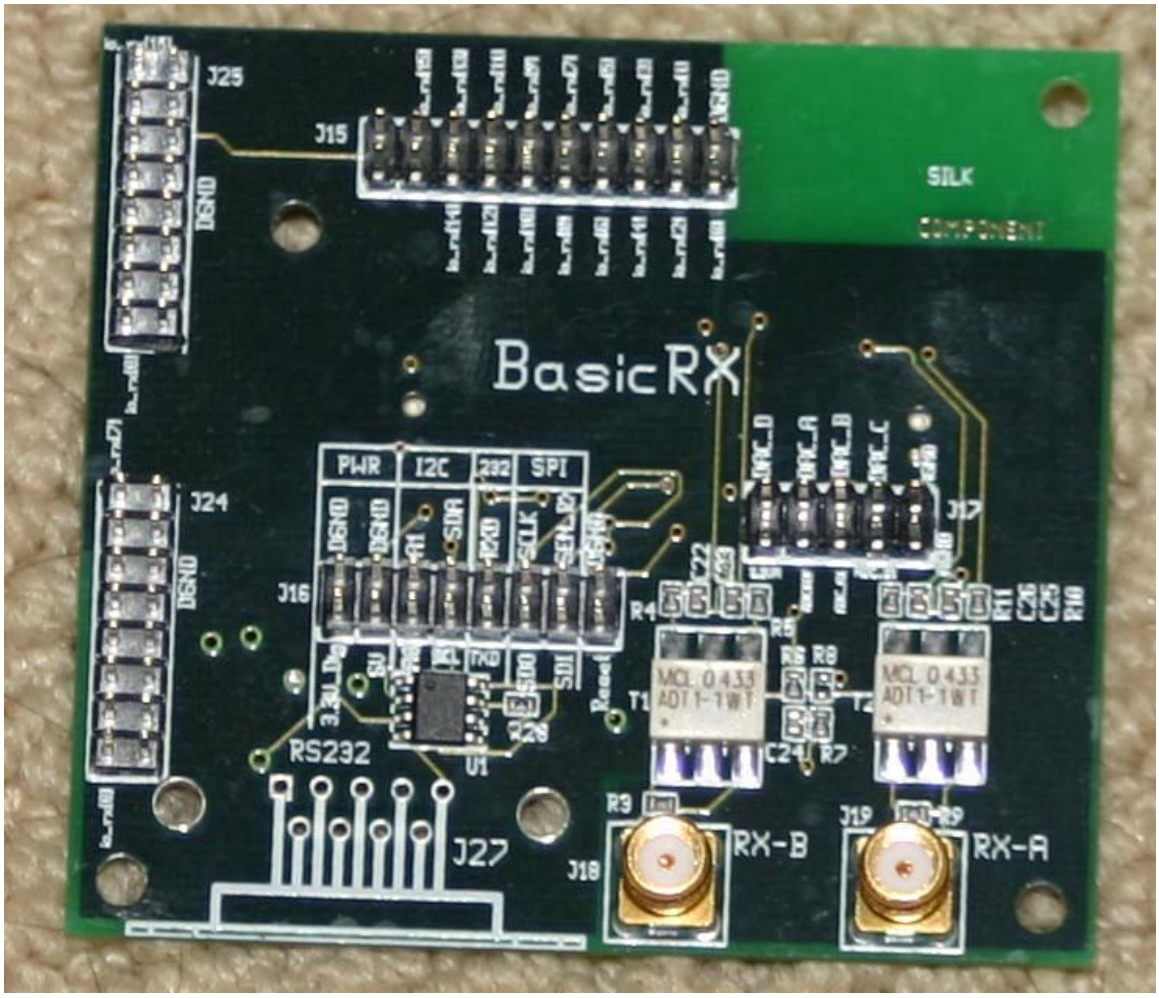

Figure 25. Basic Receive Daughterboard. (From [20]).

Utilizing the various daughterboards, the USRP has an overall range of DC to 5.9 GHz, making it a very useful tool for the radio frequency spectrum [20]. The USRP works with GNU radio, an open-source framework for the creation of software-defined radios [3].

1. USRP Receive Capabilities

The USRP daughterboard is responsible for transmitting the separated I and Q channels of the incoming complex signal to the motherboard and converting the received real signal to a complex signal at an intermediate frequency. The USRP contains four high-speed 12-bit analog-to-digital converters with sampling rates of up to 64×10^6

samples/second [20]. This gives a theoretical upper frequency limit of 32 MHz for the intermediate frequency before aliasing artifacts are introduced [22]. Two ADCs are required to convert the complex analog signal to a digital signal. One ADC is required for the I channel, and one ADC is required for the Q channel. The full voltage range of the analog-to-digital converters is 2 volts peak-to-peak, and the input is 50 ohms differential. This translates to 40 mW, or 16 dBm [22]. In the case of a weak received signal, the daughterboards can utilize programmable gain amplifiers before the ADCs to amplify the input signal up to 20 dB. The USRP also contains four digital-to-analog converters for transmission purposes, but as noted before, the transmission capabilities of the USRP were not required for this research [20].

After sampling, the motherboard FPGA processes the samples, essentially performing high bandwidth mathematical operations to reduce the data rate to an amount that is compatible with USB2.0 technology [22]. The standard FPGA configuration consists of digital down converters combined with low-pass filters. The decimation process is accomplished by low-pass filtering followed by downsampling. The USRP performs the filtering by means of cascaded integrator-comb filters [22].

Aliasing is a condition that can possibly result from digital sampling. It occurs because frequencies that differ by a multiple of the sampling rate are indistinguishable from each other after sampling [23]. The relationship between the angular analog frequency Ω_0 and the angular digital frequency ω_0 is

$$\begin{aligned} \Omega_0 &= 2\pi f_c \\ \omega_0 &= \Omega_0 t_s = 2\pi \frac{f_c}{f_s}. \end{aligned} \tag{4.1}$$

In the above equations, f_c is the frequency of the analog signal, and f_s is the sampling frequency. One can clearly see that the digital frequency is dependent on the sampling frequency. When the FPGA performs the decimation, it is effectively lowering the sampling frequency, which changes the digital frequency. Within the digital frequency domain, any frequency ω_1 and ω_2 are indistinguishable from each other given that $\omega_2 = \omega_1 + k2\pi$, where k is any integer [10]. Frequencies above the Nyquist rate are

indistinguishable from the frequencies below it, which is the basis for aliasing. Because the Nyquist frequency, which is defined as one-half of the sampling rate, corresponds to a digital frequency of π, a digital signal's bandwidth must be entirely within the range $[-\pi, \pi]$ to prevent aliasing [23]. Using (4.1), it follows that f_c must be no greater than $\frac{f_s}{2}$ to ensure aliasing will not occur.

2. USRP Transfer Capabilities

The theoretical maximum data transfer rate of a USB 2.0 connection is 480 Megabit/second [24]. In practice, however, this theoretical rate cannot be achieved; a more practical rate to be considered is approximately 256 Megabit/sec or 32 Megabytes/sec [10]. Because each complex sample that is transferred from the USRP to the personal computer is 16 bits per channel (I and Q), the total size of a sample is 32 bits. Thus, the practical sample rate utilized in this research is:

$$R_s = 256 \text{ x } 10^6 \frac{\text{bits}}{\text{sec}} \text{ x} \frac{1\text{sample}}{32\,\text{bits}}$$
$$R_s = 8 \text{ x } 10^6 \frac{\text{samples}}{\text{sec}} \tag{4.2}$$

Utilizing a lower sample rate can prove more beneficial for computer processing as it will require less storage. This will be detailed in the next section.

D. SOFTWARE SOLUTION EXPERIMENTATION

After the capabilities of GNU radio and the associated USRP were known, the next step in the research process was to integrate them through programming to demonstrate the theory developed in Chapter II. Unlike in the analog solution, only one signal generator was required for this experiment. This is due to the fact that the single signal generator can be used to simulate the bridge-to-bridge signal. A local oscillator is not used in this solution, eliminating the need for the second signal generator utilized in the analog solution. To simulate the bridge-to-bridge signal arriving at the two shipboard antennas, a phase shifter was utilized to implement a phase shift into the transmission. To simplify the experiment and to provide consistency between the hardware and

software solutions, the same Sage Laboratories® Model 6708 phase shifter [16] was used in the software solution as was used in the hardware solution. By using the same phase shifter, equivalent test conditions could be established for both the hardware and software solutions. Thus, using the same components in both experiments will allow for more valid comparisons with the results. A block diagram of the software solution experiment is shown in Figure 26.

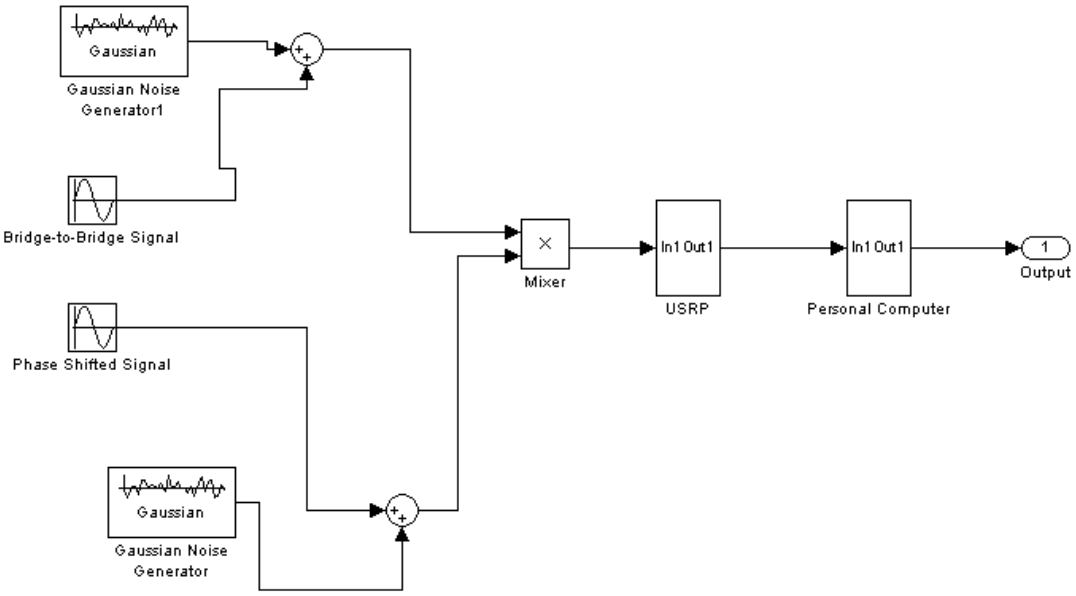

Figure 26. Software Solution Experiment Block Diagram.

For the experiment conducted with the software-defined radio, the output of the signal generator and its phase-shifted version were passed through an analog mixer before entering the receive daughterboard of the USRP. Ideally, the mixing would be performed in software along with all signal processing. Time constraints prevented the implementation of the mixer in software, but that is recommended for further research on this topic. Once, the output of the mixer entered the USRP, a program was needed to acquire the data and convert it to a format that could be analyzed and yield constructive results. The signal entering the USRP should ideally first be passed through a bandpass

filter, which would enable selection of a specific channel in the bridge-to-bridge frequency spectrum. Time constraints prevented implementation of this filter, but further work into this filter implementation is highly recommended.

As stated earlier, the GNU Radio Project contains numerous libraries for modulation, error-correcting codes, and signal processing. Along with the various blocks, the software download includes sample programs that can be used or modified to suit the user's purpose. That is a major advantage to open-source software-defined radio. By letting many users use, change, and improve the software, the project has the ability to grow at a fast rate and more economically than proprietary competitors. The sample program that I used as the base for my signal capturing program is titled *usrp_rx_cfile.py*. This program, written in the python programming language, reads samples from the USRP and then saves the data into a user-specified file format. The user is able to specify many different factors into how the program will capture the data from the USRP.

The first option lets the user select from which of the two daughterboards to collect data. As stated earlier, the USRP is able to contain up to two received daughterboards and up to two transmit daughterboards. There was only one daughterboard on the USRP used for this research, so this program specification was clear. This research utilized the BasicRX daughterboard [25] in the motherboard's "RXA" position. However, situations could arise where both daughterboards are receiving data, but the user only wishes to analyze the data from one daughterboard, but not the other. One is also able to set a limit on the number of samples to collect, any desired gain, as well as an option to produce 8-bit samples instead of the traditional 16-bit samples. The command line specification that had the greatest impact on this research and required the most experimentation was the desired decimation. One would normally desire that the decimation rate be as small as possible to obtain as many data samples as possible during a given time period. However, in the shipboard environment envisioned by this research, the relative phase change between vessels changes slowly. This will allow the user the ability to use a higher decimation rate. As shown in (4.2), the maximum sample rate for the USB2.0 connection was $8 \text{ x } 10^6 \frac{\text{samples}}{\text{sec}}$. Because the

analog-to-digital converters in the USRP operate at $64 \text{ x } 10^6 \frac{\text{samples}}{\text{sec}}$, this corresponds to a decimation rate of

$$R_{dec} = \frac{R_{USRP}}{R_{USB}}$$
$$R_{dec} = \frac{64 \text{ x} 10^6}{8 \text{ x} 10^6} = 8 \tag{4.3}$$

Thus, eight is the minimum decimation rate possible for the USB2.0 connection. If the decimation rate were less than eight, the data rate would be too high for the USB2.0 to transfer to the personal computer [24]. Additionally, storage requirements of the personal computer played in role in determining the proper decimation rate for the experiment. Because each complex sample contains 32 bits of data (16 each for the I and Q channels), the storage requirement for just one second of data collection is shown below [22]:

$$\text{Storage} = \left(\frac{32\text{bits}}{\text{sample}} \right) \left(8 \text{ x } 10^6 \frac{\text{samples}}{\text{sec}} \right)$$
$$\text{Storage} = 256 \text{ x } 10^6 \frac{\text{bits}}{\text{sec}} = 32 \text{ x } 10^6 \frac{\text{bytes}}{\text{sec}} \tag{4.4}$$
$$\text{Storage} = 32 \frac{\text{MB}}{\text{sec}}$$

This is a substantial amount, especially considering that the personal computer has purposes other than just storing data. It must have enough memory for the signal processing software, as well as its other operating requirements. In a high traffic shipping environment, the shipboard user may want to use the program numerous times in a short amount of time. Even if the ship is only tracking one vessel, it will want to use the direction-finding system on a periodic basis, regularly updating its position. Thus, the storage requirement was a major design specification.

The chosen decimated rate was also affected by the USB connection. As shown in (4.2), the maximum sample rate for the USB2.0 connection was calculated to be $8 \text{ x } 10^6 \frac{\text{samples}}{\text{sec}}$. USB2.0 technology has incorporated buffers into its protocol. These buffers are intended to "compensate for a difference in data rates or time of occurrence of events, when transmitting data from one device to another [24]." However, if the decimation rate was too low, the system experienced overflows of the USB buffer. Experimentation was conducted to determine the effect of different decimation rates. A system trade-off had to be conducted in order to choose between a low decimation rate giving a greater number of samples versus a high decimation rate which was preferable for storage and USB buffer requirements. Ultimately, a decimation rate of 16 was chosen, for it was the lowest decimation rate which yielded coherent results. This lowered the USB sampling rate to:

$$R_{USB} = \frac{64 \text{ x} 10^6}{16} = 4 \text{ x} 10^6 \frac{\text{samples}}{\text{sec}} \tag{4.5}$$

as well as lowering the storage requirement to:

$$\begin{aligned} \text{Storage} &= \left(\frac{32\text{bits}}{\text{sample}} \right) \left(4 \text{ x } 10^6 \frac{\text{samples}}{\text{sec}} \right) \\ \text{Storage} &= 128 \text{ x } 10^6 \frac{\text{bits}}{\text{sec}} = 16 \text{ x } 10^6 \frac{\text{bytes}}{\text{sec}} \\ \text{Storage} &= 16 \frac{\text{MB}}{\text{sec}} \end{aligned} \tag{4.6}$$

Because digital sampling was performed, there was the possibility of introducing the effects of aliasing. Within the software program, low-pass filtering is performed to eliminate the $\left(4\pi f_c\right)$ components, which corresponds to $2 \times 156.8 = 313.6$ MHz. This frequency is aliased to 1.6 MHz using the sample rate indicated in equation (4.5). As shown in (2.4), the output is a function of the phase difference and essentially a DC signal. Because of this property, the digital low-pass filter was able to utilize a very narrow passband. The narrow passband ensured that any aliasing introduced by the

digital sampling would be filtered out and would not affect the final results. In theory, a higher decimation rate could be utilized in the implementation of this system. It would effectively lower the sampling rate, as well as the storage rate.

With the decimation rate chosen, the usrp_rx_cfile.py program could be utilized to analyze the direction-finding system. This program captured samples and saved them into a *.dat formatted file. The data stored in this format could then be read and analyzed to determine the performance of the direction-finding system. As stated earlier, the phase shifter was used to implement the phase difference that would occur between a single signal arriving at two separate antennas. The phase shifter was capable of implementing phase changes from $[0,180^{\circ}]$. It was determined that 30 turns were necessary to achieve the full 180 degrees of phase change; thus, each turn of the phase shifter implemented six degrees of phase change. To establish a baseline for the system, the program was run thirty times to capture data for each turn of the phase shifter. The program was run for one second in order to capture an adequate amount of data and then saved in a *.dat formatted file. Once the data was captured, it needed to be analyzed to determine the differences in output caused by each turn of the phase shifter.

The data analysis software chosen for this research was GNU Octave. GNU octave is a high-level language intended for numerical calculation. It provides a convenient command line interface using a language that is strikingly similar to MATLAB [4]. Octave, like GNU radio, is open-source software that is free to be used and improved upon by any user. It can be utilized with functions in Octave's own language, which was done in this research, but can also be customized with modules written in C++, C, Fortran, or other languages [4].

The first step in establishing the baseline for the direction-finding system was to convert the data captured by the usrp_rx_cfile.py program into readable, analyzable data. This was accomplished in Octave by using the function read_complex_binary.m. This program opens the file specified and returns its contents as a column vector of 32 bit complex numbers. In research conducted before the experiment was performed, it was discovered that the USRP occasionally has issues with a buffer contained within the

FPGA. The buffer is not "flushed" as it should be when the USRP is opened or closed. This distorts the beginning samples of each set of data collected by usrp_rx_cfile.py. The open-source community is current investigating solutions to this problem, but for this research, a work-around was achieved. Because of the large amount of data collected in one second, it was possible to drop the first one million samples to remove the adverse affects of the non-flushed FPGA without losing too much data to make the results unusable. Dropping this data yielded more coherent solutions for the direction-finding system.

After the data was resized, the program calculated the magnitude of the signal by using the formula

$$\text{Magnitude}=\sqrt{I^2+Q^2}. \tag{4.7}$$

A low-pass filter was created in Octave to filter out the high-frequency components that resulted from the analog mixing before entry into the USRP. After filtering and according to (2.8), the output of the filter should be a function of the phase difference of the two input signals. Within the octave program, the mean value of the filter output was calculated in order to determine if there was a recognizable function that resulted from plotting the output of the program versus the phase difference. Figure 27 shows the baseline established for the direction-finding system.

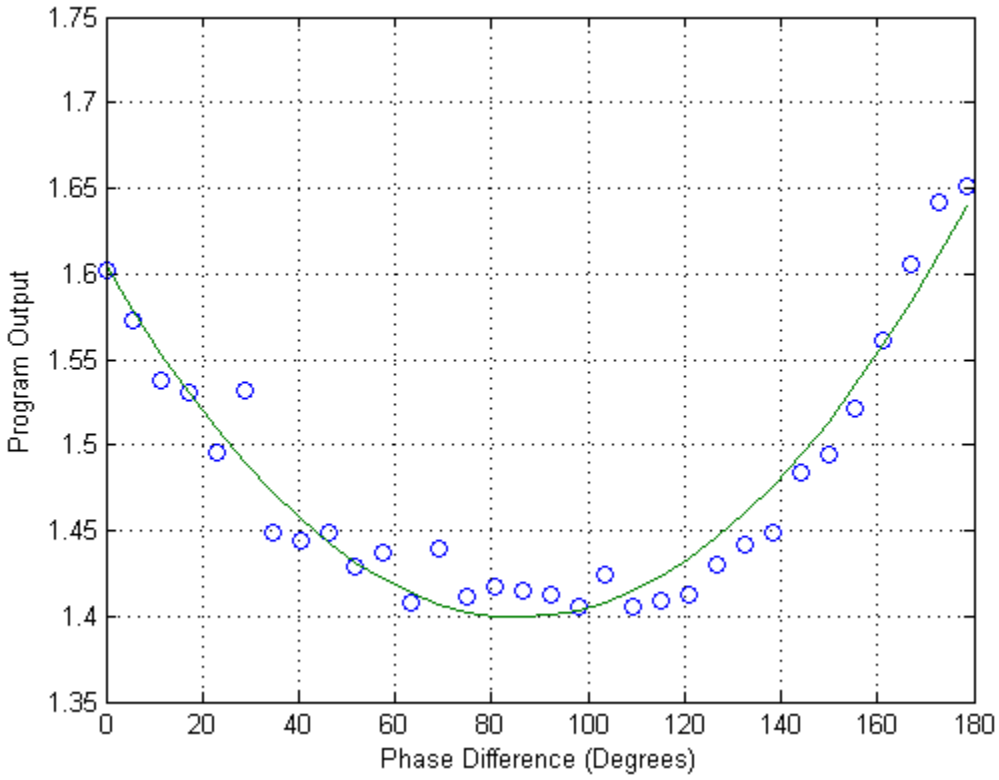

Figure 27. Relationship Between Phase Difference and Program Output.

The experimental data is shown in circles with a best-fit curve plotted on top of the data. As with the hardware solution, one does not see the expected trigonometric relationship. While the hardware solution yielded a linear relationship, the software solution yields a nonlinear, parabolic relationship. It is unclear why the hardware and software results differ in shape, but time constraints prevented further investigation into this difference. One can clearly see this relationship between the phase difference and the program output. Using the equation for the best-fit line, the relationship between phase and program output is shown below:

$$\text{phase(deg)} = 189.1\left(\sqrt{\text{output} - 1.4} + 0.454\right)$$
$$\text{or} \qquad (4.8)$$
$$\text{phase(deg)} = -189.1\left(\sqrt{\text{output} - 1.4} - 0.454\right)$$

Because of the above relationships, there are two solutions for every output value, which is a weakness of the software solution as compared with the hardware solution.

This phase information is then entered into another Octave program which uses (2.4), derived in Chapter II, to arrive at a solution set of possible Angle of Arrivals for Antenna System 1. Due to the parabolic shape seen in Figure 26, each output from the program can lead to two possible phase angles. As shown in the previous chapter, each phase leads to two possible bearings. Thus, with the software solution, there are four possible bearings, even when using a second antenna system. This is worse than the hardware solution, for there is twice as much bearing ambiguity.

Thus, the software solution has the capability of solving the problem for which the research was conducted. It has proved that a functioning system can be built with predominantly software components that can accurately determine the Angle of Arrival of an incoming bridge-to-bridge signal if the phase difference between a single signal arriving at different locations is known.

THIS PAGE INTENTIONALLY LEFT BLANK

V. CONCLUSION

A. CONCLUSIONS

1. Performance Conclusions

This research has shown that it is possible to perform direction finding of a bridge-to-bridge VHF band signal using both traditional hardware components as well as software-defined radio. Common techniques for conducting geolocation focused on Frequency Difference of Arrival and Time Difference of Arrival. This research confirmed that in addition to the above techniques, interferometry based on phase difference is a powerful tool that can also be utilized. While the software-based solution has the most room for future growth, the scope of this research has shown the analog solution to be the preferred option. Both the hardware and software solutions were unable to develop a single Angle of Arrival solution due to the multivalued nature of the arcsine and arccosine. However, the linear nature of the hardware solution and its tighter best-fit line yield a more reliable, preferable solution. Within the range[0,180], there is a unique solution for each phase. However, the parabolic shape of the software solution has the possibility of two phases for each output of the software system. In both these cases, the addition of a second antenna system greatly aids performance. Even in this scenario, however, the hardware solution still outperforms the software solution. For as stated in the previous chapter, the hardware solution yields two possible bearings for the source of the bridge-to-bridge signal, but the software solution yields four possible bearings. Thus, the bearing ambiguity in the software solution is worse than the hardware solution.

There is limited room for improvement in the hardware solution. All components used in the experiment were top-of-the-line, standard components used in commercial applications. However, the hardware used in the software solution could be improved. While still more expensive than the hardware solution, the USRP is considered a low-cost device. However, it is not the only platform that can be used for software-defined radio. It is popular due to its affordability and flexibility. If this system were to be developed, a

more specialized hardware may be desired. If one is solely concerned about the bridge-to-bridge frequency range or other VHF signals, one would not need hardware that has as large a frequency range as the USRP. The USRP has capability up to the gigahertz range, which would not be needed in a specialized system. The specialized system could also utilize a much lower sample rate. Due to the slow-changing nature of the phase between two maritime vessels, the high sample rate is not a mandatory requirement. Although it would likely add cost, performance could likely be improved if hardware were utilized that was specially tailored for the specific, desired frequency range.

2. Cost Conclusions

In the design of any engineering system, the performance of the system is only one consideration in the decision for further development, procurement, and production. It is a very important factor, but in the majority of situations, the lifecycle cost is the most important factor. One may achieve unmatched performance, but if it is extremely expensive throughout the system lifecycle it will not be produced and development may be stopped. On the other hand, the cheapest solution may not achieve acceptable performance. A trade-off study must often be conducted to achieve the optimal balance between cost and performance. The cost associated with the hardware solution explored in this research breaks down as follows:

- $220.22 for circuit board components
- $130.00 for two circuit boards (one per antenna system)

This does not include the test equipment utilized in the Naval Postgraduate School laboratory, which included the phase shifter, two signal generators, and many connecting wires and cables. The cost required for the software solution is as follows (from [25]):

- $700.00 for USRP package
- $75.00 per daughterboard

Based on both cost and performance in the experiments conducted, the hardware solution is the preferable choice when compared to the software solution. However, the software

solution has the most room for growth. The results of future work and experimentation may improve the performance of the software solution to the point where it may be worth the extra cost.

B. RECOMMENDATIONS

There is significant room for future work and experimentation with both the software and hardware solutions. For the software solution, future work should be undertaken to implement more of the functionality in software. Time constraints precluded implementing the signal mixing in software. Additionally, a bandpass filter should be implemented in software and/or hardware to ensure that the signals that enter into the digital mixer are the frequency desired. The software experiment conducted used a signal generator to ensure just one frequency was analyzed by the system. However, in a real-world scenario, there will be many signals of many different frequencies being transmitted. The software would need the bandpass filter to ensure that only frequencies of interest are captured and analyzed. Further research could also be conducted into conducting the software solution experiment differently. Instead of using a decimation factor to lower the USB sampling rate, it could be possible to divide the information from the USRP into manageably-sized blocks of data. This data could then be transferred to the personal computer without having to sample, which would ensure that no data samples are excluded during analysis.

The local oscillator addition to the hardware solution ensured that this problem was accounted for in the hardware solution. Thus, the hardware solution is farther along in the development process than the software solution. However, future work remains for the hardware solution. The next step in research would be to test both systems with wireless signals. Using hard-wired signal generators to simulate the bridge-to-bridge signals minimizes the loss that would be inherent in real-world signals. The noise and interference that would be experienced with real-world signals needs to be taken into account. This most likely would require the addition of a Low Noise Amplifier to both the hardware and software solutions. Automatic Gain Control would also most likely need to be implemented to ensure a consistent input voltage and power level to the

system. The wireless system should first be tested in a controlled environment, such as an anechoic chamber to shield the RF waves. After successful testing in the controlled environment, testing should be conducted in a real-world environment. Finally, full-shipboard testing should occur in a maritime environment for a final demonstration of the direction-finding system.

The potential exists for additions to the system that would make it even more powerful to the user. Further investigation could be conducted to determine the precise reason why the hardware and software solutions yielded such different results. A display could be designed that would convert the calculated Angles of Arrival into a graphical display. In the current hardware system, the voltage output is supplied to a MATLAB program to calculate the Angle of Arrival. A device could be designed and built that used the voltage output and calculated the bearings without any user input. Further research should also be conducted to resolve the ambiguity problem. Possible solutions may include adding another set of antennas or perhaps comparing the two antenna signals in a different way that would yield only one line of bearing. To convert the results of this research to a final, finished product would require the testing detailed in the previous paragraph. It would also require the integration of the system built in this research, with a microprocessor that performs all calculations and a textual or graphical display for enhanced usability.

LIST OF REFERENCES

[1] "U.S. VHF Channels." http://www.navcen.uscg.gov/marcomms/vhf.htm. Accessed 10 February 2009.

[2] Martin Benedyk. "Navy Officials Say Threat Raised Tension." http://www.foxnews.com/wires/2008Jan13/0,4670,IranUSNavy,00.html. Accessed 03 February 2009.

[3] "GNU Radio Introduction." http://gnuradio.org/trac. Accessed 19 February 2009.

[4] "About Octave." http://www.gnu.org/software/octave/about.html. Accessed 23 February 2009.

[5] Derek Elsaesser. "The Discrete Probability Density Method for Emitter Geolocation." Canadian Conference on Electrical and Computer Engineering, 2006, pp. 25-30.

[6] A. Mikhalev and R.F. Ormondroyd. "Fusion of Sensor Data for Source Localization using the Hough Transform." 9th International Conference on Information Fusion, 2006. pp. 1-6.

[7] A. Mikhalev and R.F. Ormondroyd. "Comparison of Hough Transform and Particle Filter Methods of Emitter Geolocation using Fusion of TDOA Data." 4th Workshop on Positioning, Navigation, and Communication 2007. pp. 121-127.

[8] Alex Mikhalev and Richard Ormondroyd. "Passive Emitter Geolocation using Agent-based Data Fusion of AOA, TDOA and FDOA Measurements. 10th International Conference on Information Fusion, 2007. pp. 1-6.

[9] David Adamy. *EW101: A First Course in Electronic Warfare*. Artech House Publishers, Boston, 2001.

[10] Ian Larsen. Design and Implementation of a Mobile Phone Locator Using Software Defined Radio. Master's Thesis, Naval Postgraduate School, 2007.

[11] "Introduction to Simulink." http://www.mathworks.com/products/simulink/description1.html. Accessed 08 February 2009.

[12] "Elsie." http://www.tonnesoftware.com/elsie.html. Accessed 16 February 2009.

[13] Jeffrey S. Beasley and Gary M. Miller. *Modern Electronic Communication, Eighth Edition.* Pearson Prentice Hall, Upper Saddle River, 2005.

[14] "The ARRL Handbook for Radio Communications." ARRL-The National Association for Amateur Radio. Newington, CT. 2004.

[15] PCB123. http://www.pcb123.com/. Accessed 27 February 2009.

[16] Sage Laboratories, Inc. http://www.sagelabs.com/comp/phase_shifters.shtml. Accessed 27 February 2009.

[17] Mini-Circuits ZLW-1 Frequency Mixer. http://www.minicircuits.com/pdfs/ZLW-1.pdf. Accessed 27 February 2009.

[18] Hewlett Packard 8341B Synthesized Sweeper. http://www.teknetelectronics.com/DataSheet/HP_AGILENT/HP__8341B.pdf. Accessed 27 February 2009.

[19] "Benefits of SDR." http://www.sdrforum.org/pages/aboutSdrTech/benefitsOfSdr.asp. Accessed 09 March 2009.

[20] "USRP Overview." http://www.ettus.com/downloads/er_broch_trifold_v5b.pdf. Accessed 20 February 2009.

[21] "Overview of Ossie." http://ossie.wireless.vt.edu/about.html. Accessed 17 February 2009.

[22] "The USRP Board." https://radioware.ee.nd.edu/documentation/hardware/the-usrp-board. Accessed 20 February 2009.

[23] Roberto Christi. Modern Digital Signal Processing. Brooks/Cole, 2003.

[24] "Universal Serial Bus Specification Revision 2.0." Developed by Compaq, Hewlett-Packard, Intel, Lucent, Microsoft, NEC, Philips. April 27, 2000.

[25] "USRP Order Page." http://www.ettus.com/orderpage.html. Accessed 20 February 2009.

INITIAL DISTRIBUTION LIST

1. Defense Technical Information Center
   Ft. Belvoir, VA

2. Dudley Knox Library
   Naval Postgraduate School
   Monterey, CA

3. Eric Boernke
   Naval Postgraduate School
   Monterey, CA

4. Frank Kragh
   Naval Postgraduate School
   Monterey, CA

5. Herschel Loomis
   Naval Postgraduate School
   Monterey, CA

6. Donna Miller
   Naval Postgraduate School
   Monterey, CA

7. Bob Broadston
   Naval Postgraduate School
   Monterey, CA

8. Tri Ha
   Naval Postgraduate School
   Monterey, CA

9. Tracy Conroy
   Science Advisor, COMNAVSURFOR
   San Diego, CA

10. Professor Yeo Tat Soon, Director
    Temasek Defence Systems Institute
    National University of Singapore
    Singapore

11. Tan Lai Poh (Ms), Assistant Manager
    Temasek Defence Systems Institute
    National University of Singapore
    Singapore